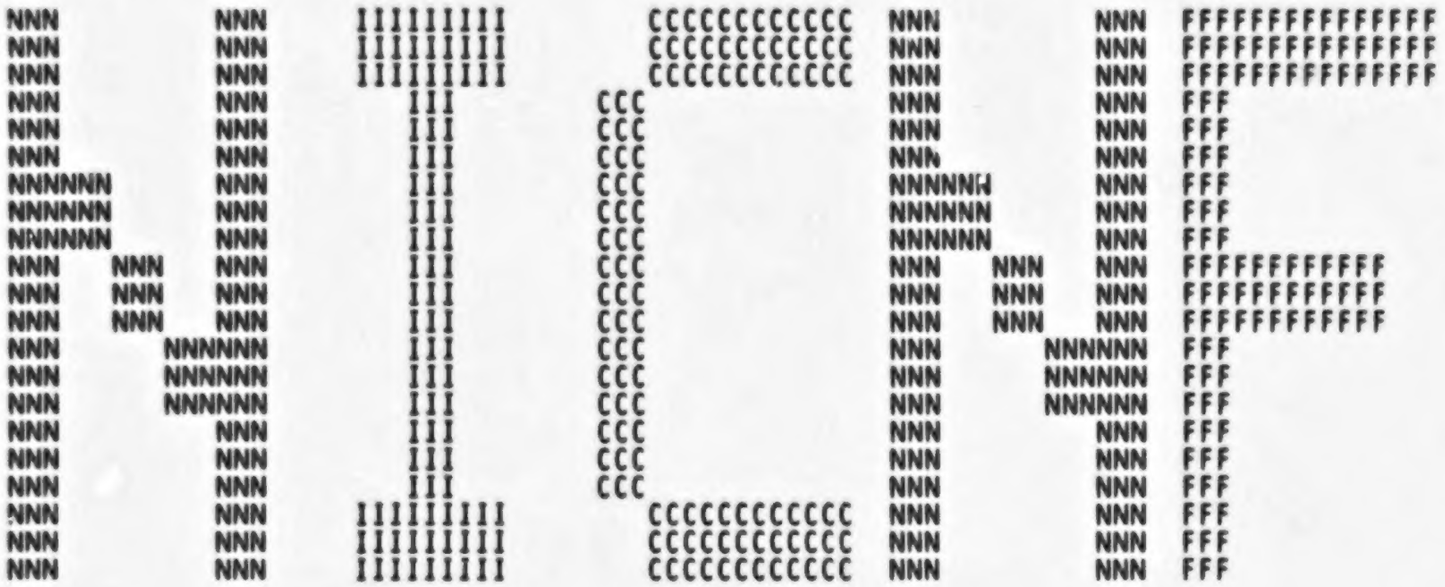
NNN NNN IIIIIIIII CCCCCCCCCCC NNN NNN FFFFFFFFFFFFFFF
NNN NNN IIIIIIIII CCCCCCCCCCC NNN NNN FFFFFFFFFFFFFFF
NNN NNN IIIIIIIII CCCCCCCCCCC NNN NNN FFFFFFFFFFFFFFF
NNN NNN III CCC NNN NNN FFF
NNN NNN III CCC NNN NNN FFF
NNN NNN III CCC NNN NNN FFF
NNNNNN NNN III CCC NNNNNN NNN FFF
NNNNNN NNN III CCC NNNNNN NNN FFF
NNNNNN NNN III CCC NNNNNN NNN FFF
NNN NNN NNN III CCC NNN NNN NNN FFFFFFFFFFF
NNN NNN NNN III CCC NNN NNN NNN FFFFFFFFFFF
NNN NNN NNN III CCC NNN NNN NNN FFFFFFFFFFF
NNN NNNNNN III CCC NNN NNNNNN FFF
NNN NNNNNN III CCC NNN NNNNNN FFF
NNN NNNNNN III CCC NNN NNNNNN FFF
NNN NNN III CCC NNN NNN FFF
NNN NNN III CCC NNN NNN FFF
NNN NNN III CCC NNN NNN FFF
NNN NNN IIIIIIIII CCCCCCCCCCC NNN NNN FFF
NNN NNN IIIIIIIII CCCCCCCCCCC NNN NNN FFF
NNN NNN IIIIIIIII CCCCCCCCCCC NNN NNN FFF

FILEID**CNFREQUES

```
;  1     0001  0 %TITLE  'DECnet Ethernet Configurator Module'
;  2     0002  0 MODULE CNFREQUES        (
;  3     0003  0                         LANGUAGE (BLISS32),
;  4     0004  0                         IDENT = 'V04-000'
;  5     0005  0                         ) =
;  6     0006  1 BEGIN
;  7     0007  1
;  8     0008  1 !
;  9     0009  1 !*****************************************************************************
; 10     0010  1 !*                                                                           *
; 11     0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 12     0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
; 13     0013  1 !*  ALL RIGHTS RESERVED.                                                      *
; 14     0014  1 !*                                                                           *
; 15     0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 16     0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 17     0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 18     0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 19     0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 20     0020  1 !*  TRANSFERRED.                                                             *
; 21     0021  1 !*                                                                           *
; 22     0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 23     0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 24     0024  1 !*  CORPORATION.                                                             *
; 25     0025  1 !*                                                                           *
; 26     0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 27     0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 28     0028  1 !*                                                                           *
; 29     0029  1 !*                                                                           *
; 30     0030  1 !*****************************************************************************
; 31     0031  1 !
; 32     0032  1
; 33     0033  1 !++
; 34     0034  1 ! FACILITY:     DECnet Configurator Module (NICONFIG)
; 35     0035  1 !
; 36     0036  1 ! ABSTRACT:
; 37     0037  1 !
; 38     0038  1 !       This module contains the routines to process incoming requests
; 39     0039  1 !       by parsing them and dispatching to the appropriate routines.
; 40     0040  1 !
; 41     0041  1 ! ENVIRONMENT:  VAX/VMS Operating System
; 42     0042  1 !
; 43     0043  1 ! AUTHOR:       Bob Grosso,     CREATION DATE:  13-Oct-1982
; 44     0044  1 !
; 45     0045  1 ! MODIFIED BY:
; 46     0046  1 !
; 47     0047  1 !       V03-002 RPG0002         Bob Grosso              16-May-1983
; 48     0048  1 !               Correct the arguement list to a call to Signal.
; 49     0049  1 !
; 50     0050  1 !       V03-001 RPG0001         Bob Grosso              02-May-1983
; 51     0051  1 !               Check state of UNA.
; 52     0052  1 !--
```

```
;    54          0053  1 %SBTTL  'Definitions'
;    55          0054  1
;    56          0055  1 !
;    57          0056  1 ! INCLUDE FILES:
;    58          0057  1 !
;    59          0058  1
;    60          0059  1 LIBRARY 'SYS$LIBRARY:STARLET';                  ! VMS common definitions
;    61          0060  1
;    62          0061  1 LIBRARY 'SHRLIB$:NET';                          ! Network definitions
;    63          0062  1
;    64          0063  1 LIBRARY 'SHRLIB$:NMALIBRY';                     ! NICE code definitions
;    65          0064  1
;    66          0065  1 REQUIRE 'LIB$:CNFDEF.R32';
;    67          0156  1
;    68          0157  1 REQUIRE 'SRC$:CNFPREFIX.REQ';
;    69          0254  1
;    70          0255  1
;    71          0256  1 !
;    72          0257  1 ! BUILTIN functions
;    73          0258  1 !
;    74          0259  1
;    75          0260  1 BUILTIN
;    76          0261  1     INSQUE,                                  ! INSQUE instruction
;    77          0262  1     REMQUE;                                  ! REMQUE instruction
;    78          0263  1
;    79          0264  1
;    80          0265  1 !
;    81          0266  1 ! TABLE OF CONTENTS:
;    82          0267  1 !
;    83          0268  1
;    84          0269  1 FORWARD ROUTINE
;    85          0270  1
;    86          0271  1         CNF$PROCESS_REQUEST : NOVALUE,   ! Jacket routine for Process_request
;    87          0272  1         PROCESS_REQUEST,                 ! Parse NICE and dispatch
;    88          0273  1         CNF_ENABLE_SURVEILLANCE,         ! Jacket routine for enable_surveillance
;    89          0274  1         ENABLE_SURVEILLANCE,             ! Set-up to prepare for setting surveillance
;    90          0275  1         SURVEIL,                         ! Begin surveillance of a circuit
;    91          0276  1         CNF$LOCATE_CIR_BLK,              ! Match an ASCII circuit name with a CIR control block
;    92          0277  1         CNF_DISABLE_SURVEILLANCE,        ! Jacket routine for disable_surveillance
;    93          0278  1         DISABLE_SURVEILLANCE,            ! set-up to prepare to discontinue circuit surveillance
;    94          0279  1         CNF$DISABLE_SURVEIL;             ! disabled surveillance of a circuit
;    95          0280  1
;    96          0281  1
;    97          0282  1
;    98          0283  1 !
;    99          0284  1 ! EXTERNAL REFERENCES:
;   100          0285  1 !
;   101          0286  1
;   102          0287  1 EXTERNAL ROUTINE
;   103          0288  1
;   104          0289  1
;   105          0290  1 !   Module CNFMAIN
;   106          0291  1         CNF$EXIT,                        ! Clean up and exit
;   107          0292  1         CNF$TRACE,                       ! Log messages to log file
;   108          0293  1         CNF$LOG_DATA,                    ! Log messages to log file
;   109          0294  1         CNF$GET_ZVM,                     ! Get zeroed virtual memory
;   110          0295  1         CNF$FREE_VM,                     ! Free virtual memory
```

```
;  111     0296  1
;  112     0297  1 !   Module CNFSTORE
;  113     0298  1
;  114     0299  1         CNF$READ_SYSIDM,                        ! Issue QIO to listen on the NI
;  115     0300  1
;  116     0301  1 !   Module CNFSHOW
;  117     0302  1
;  118     0303  1         CNF$PROCESS_SHOW,                       ! Show Circuit and system IDs
;  119     0304  1
;  120     0305  1 !   Module CNFSEND
;  121     0306  1
;  122     0307  1         CNF$BUFR_NICE_MSG,
;  123     0308  1         CNF$BUFR_ERR_MSG,
;  124     0309  1         CNF$SEND_NICE_MSG,
;  125     0310  1
;  126     0311  1 !   Module CNFWORKQ
;  127     0312  1
;  128     0313  1         WKQ$ADD_WORK_ITEM;                      ! Add work to work queue
;  129     0314  1
;  130     0315  1 EXTERNAL ROUTINE
;  131     0316  1
;  132     0317  1         STR$COMPARE             : ADDRESSING_MODE (GENERAL);
;  133     0318  1
;  134     0319  1
;  135     0320  1 EXTERNAL LITERAL
;  136     0321  1
;  137     0322  1         CNF$_CHAN,              ! Error assigning or deassigning channel
;  138     0323  1         CNF$_DRVRSTRT,          ! Error while issuing startup command to driver
;  139     0324  1         CNF$_LOGIC,             ! Program logic error or unexpected condition
;  140     0325  1
;  141     0326  1 !   From CNFSTORE
;  142     0327  1         SYSIDM_BUFSIZ,
;  143     0328  1         ADRTYP_BUFSIZ,
;  144     0329  1
;  145     0330  1         CNF$C_SYNCH_EFN,
;  146     0331  1         CNF$C_ASYNCH_EFN;
;  147     0332  1
;  148     0333  1
;  149     0334  1 EXTERNAL
;  150     0335  1
;  151     0336  1         CNF$B_SURVEILLANCE_SET,                 ! Boolean: mark if anything is under surveillance
;  152     0337  1         CNF$W_NETCHAN : WORD,                   ! Channel opened to network
;  153     0338  1         CNF$GQ_CIRSURLST : VECTOR [2];          ! List of circuit under surveillance
;  154     0339  1
;  155     0340  1 OWN
;  156     0341  1     SUCCESS_NICE_DSC :
;  157     0342  1         BBLOCK [DSC$C_S_BLN] INITIAL
;  158     0343  1         (
;  159     0344  1             4,
;  160     0345  1             UPLIT (
;  161     0346  1                 BYTE (%X'01'),
;  162     0347  1                 WORD (%X'FFFF'),
;  163     0348  1                 BYTE (%X'00')
;  164     0349  1                 )
;  165     0350  1         );
```

```
;  167          0351  1 %SBTTL  'cnf$process_request'
;  168          0352  1 GLOBAL ROUTINE CNF$PROCESS_REQUEST (IRB) : NOVALUE =
;  169          0353  1
;  170          0354  1 !++
;  171          0355  1 ! FUNCTIONAL DESCRIPTION:
;  172          0356  1 !
;  173          0357  1 !  This routine is executed off the work queue.
;  174          0358  1 !  Parse the NICE message to determine the type of operation,
;  175          0359  1 !  and the circuits to be affected.  Dispatch to appropriate
;  176          0360  1 !  routine if entire message is correct.
;  177          0361  1 !
;  178          0362  1 ! FORMAL PARAMETERS:
;  179          0363  1 !
;  180          0364  1 !       irb     Interrupt request block, contains all the info for a connection
;  181          0365  1 !               to NICONFIG.  The IRB contains the NICE command which will
;  182          0366  1 !               be parsed.
;  183          0367  1 !
;  184          0368  1 ! IMPLICIT INPUTS:
;  185          0369  1 !       NONE
;  186          0370  1 !
;  187          0371  1 ! IMPLICIT OUTPUTS:
;  188          0372  1 !       NONE
;  189          0373  1 !
;  190          0374  1 ! ROUTINE VALUE:
;  191          0375  1 ! COMPLETION CODES:
;  192          0376  1 !       Success
;  193          0377  1 !
;  194          0378  1 ! SIDE EFFECTS:
;  195          0379  1 !       NONE
;  196          0380  1 !
;  197          0381  1 !--
;  198          0382  1
;  199          0383  2     BEGIN
;  200          0384  2     LOCAL
;  201          0385  2         CIRCUIT_DSC : BBLOCK [DSC$C_S_BLN],      ! Allocate circuit name descriptor here, whether it will be
;  202          0386  2                                                  !  or not, it makes book keeping much simpler.
;  203          0387  2         STATUS;
;  204          0388  2
;  205          0389  2     CH$FILL (0, DSC$C_S_BLN, CIRCUIT_DSC);               ! Zero the descriptor
;  206          0390  2     STATUS = PROCESS_REQUEST (.IRB, CIRCUIT_DSC);        ! Parse and act upon the command
;  207          0391  2     IF NOT .STATUS                                       ! If unsuccessful, buffer an error message for retur
;  208          0392  2     THEN
;  209          0393  2         CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_RES, 0, .STATUS, 0);
;  210          0394  2
;  211          0395  2     CNF$SEND_NICE_MSG (.IRB);                            ! Issue QIO's to send NICE messages buffered
;  212          0396  2
;  213          0397  2     IF .CIRCUIT_DSC [DSC$W_LENGTH] NEQ 0                 ! If a buffer was allocated to the descriptor, retur
;  214          0398  2     THEN
;  215          0399  2         CNF$FREE_VM (CIRCUIT_DSC [DSC$W_LENGTH], CIRCUIT_DSC [DSC$A_POINTER]);
;  216          0400  2
;  217          0401  2     RETURN TRUE;                                         ! Always return success, errors are sent via QIO bac
;  218          0402  1     END;                                                 ! Routine cnf$process_request


                                                        .TITLE  CNFREQUES DECnet Ethernet Configurator Module
                                                        .IDENT  \V04-000\
```

```
                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                              01  00000 P.AAA:  .BYTE   1                                                ;
                            FFFF  00001         .WORD   -1                                               ;
                              00  00003         .BYTE   0                                                ;

                                        .PSECT  $OWN$,NOEXE,2

                        00000004  00000 SUCCESS_NICE_DSC:
                                        .LONG   4                                                        ;
                        00000000' 00004         .ADDRESS P.AAA                                           ;

                                        .EXTRN  CNF$EXIT, CNF$TRACE
                                        .EXTRN  CNF$LOG_DATA, CNF$GET_ZVM
                                        .EXTRN  CNF$FREE_VM, CNF$READ_SYSIDM
                                        .EXTRN  CNF$PROCESS_SHOW
                                        .EXTRN  CNF$BUFR_NICE_MSG
                                        .EXTRN  CNF$BUFR_ERR_MSG
                                        .EXTRN  CNF$SEND_NICE_MSG
                                        .EXTRN  WKQ$ADD_WORK_ITEM
                                        .EXTRN  STR$COMPARE, CNF$_CHAN
                                        .EXTRN  CNF$_DRVRSTRT, CNF$_LOGIC
                                        .EXTRN  SYSIDM_BUFSIZ, ADRTYP_BUFSIZ
                                        .EXTRN  CNF$C_SYNCH_EFN
                                        .EXTRN  CNF$C_ASYNCH_EFN
                                        .EXTRN  CNF$B_SURVEILLANCE_SET
                                        .EXTRN  CNF$W_NETCHAN, CNF$GQ_CIRSURLST

                                        .PSECT  $CODE$,NOWRT,2

                                  003C  00000         .ENTRY  CNF$PROCESS_REQUEST, Save R2,R3,R4,R5    ; 0352
                              5E        00002         SUBL2   #8, SP                                   ;
        08          00        6E  08  00005         MOVC5   #0, (SP), #0, #8, CIRCUIT_DSC            ; 0389
                              6E      0000A                                                            ;
                              5E  DD  0000B         PUSHL   SP                                       ; 0390
                          04  AC  DD  0000D         PUSHL   IRB                                      ;
                    0000V     CF  02  00010         CALLS   #2, PROCESS_REQUEST                      ;
                              11  50  00015         BLBS    STATUS, 1$                               ; 0391
                                  7E  00018         CLRL    -(SP)                                    ; 0393
                                  50  0001A         PUSHL   STATUS                                   ;
                                  7E  0001C         CLRL    -(SP)                                    ;
                              7E  0F  0001E         MNEGL   #15, -(SP)                               ;
                          04  AC  DD  00021         PUSHL   IRB                                      ;
                    0000G     CF  05  00024         CALLS   #5, CNF$BUFR_ERR_MSG                     ;
                          04  AC  DD  00029 1$:     PUSHL   IRB                                      ; 0395
                    0000G     CF  01  0002C         CALLS   #1, CNF$SEND_NICE_MSG                    ;
                                  6E  B5  00031     TSTW    CIRCUIT_DSC                              ; 0397
                                  0B  13  00033     BEQL    2$                                       ;
                          04  AE  9F  00035         PUSHAB  CIRCUIT_DSC+4                            ; 0399
                          04  AE  9F  00038         PUSHAB  CIRCUIT_DSC                              ;
                    0000G     CF  02  0003B         CALLS   #2, CNF$FREE_VM                          ;
                                  04  00040 2$:     RET                                              ; 0402

; Routine Size:  65 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  220     0403  1 %SBTTL  'process_request'
;  221     0404  1 ROUTINE PROCESS_REQUEST (IRB, CIRNAM_DSC) =
;  222     0405  1
;  223     0406  1 !++
;  224     0407  1 !
;  225     0408  1 !  This routine is called by CNF$PROCESS_REQUEST which is
;  226     0409  1 !  executed off the work queue.
;  227     0410  1 !  Parse the NICE message to determine the type of operation,
;  228     0411  1 !  and the circuits to be affected.  Dispatch to appropriate
;  229     0412  1 !  routine if entire message is correct.
;  230     0413  1 !
;  231     0414  1 !
;  232     0415  1 !       irb             Interrupt request block, contains all the info for
;  233     0416  1 !                       a connection to NICONFIG.  The IRB contains the
;  234     0417  1 !                       NICE command which will be parsed.
;  235     0418  1 !
;  236     0419  1 !       cirnam_dsc      Descriptor for storing of circuit name if one is
;  237     0420  1 !                       specified in command.
;  238     0421  1 !
;  239     0422  1 !--
;  240     0423  1
;  241     0424  2     BEGIN
;  242     0425  2     MAP
;  243     0426  2         CIRNAM_DSC : REF BBLOCK [DSC$C_S_BLN],
;  244     0427  2         IRB : REF BBLOCK;                   ! Interrupt request block
;  245     0428  2
;  246     0429  2     LOCAL
;  247     0430  2         KNOWN,                              ! Was KNOWN CIRCUITS present in command
;  248     0431  2         NICE : REF BBLOCK,                  ! Pointer into NICE command
;  249     0432  2         FUNCTION : BBLOCK [1],
;  250     0433  2         OPTION : BBLOCK [1],
;  251     0434  2         PROCESSING_SHOW,                    ! Boolean, true = SHOW, false = SET
;  252     0435  2         SHOW_INFO,                          ! Coded for CHAR, SUMMARY or STATUS
;  253     0436  2         LEN_REMAINING,
;  254     0437  2         NICE_SURVEILLANCE : REF BBLOCK; ! Locate section of NICE command
;  255     0438  2                                         !  containing the SURVEILLANCE parameter
;  256     0439  2
;  257     0440  2     BIND
;  258     0441  2         CONF = UPLIT (%ASCIC 'CONFIGURATOR') : VECTOR [,BYTE];
;  259     0442  2
;  260     0443  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;  261     0444  2         $DESCRIPTOR ('process_request'));
;  262     0445  2
;  263     0446  2     NICE = IRB [IRB$T_REQUEST];                 ! Beginning of NICE command
;  264     0447  2
;  265     0448  2     IF .IRB [IRB$W_IOSB1]                       ! The size of the NICE message was returned in the IOSB
;  266     0449  2         LSS 18                                  ! NICE message too short to contain
;  267     0450  2     THEN                                        ! function, option, and "CONFIGURATOR", and Circuit
;  268     0451  3         BEGIN
;  269     0452  3         CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_INV, 0, 0, 0);
;  270     0453  3         RETURN TRUE;
;  271     0454  3         END;
;  272     0455  2
;  273     0456  2 !
;  274     0457  2 !       An Acceptable NICE message must conform with the following
;  275     0458  2 !
;  276     0459  2 !   Byte 1      Function byte, accept either CHANGE or READ
```

```
;  277     0460  2 !   Byte 2      Option Byte,
;  278     0461  2 !                 bits 0-2 contain the entity type. Accept only MODULE
;  279     0462  2 !                 For Function READ
;  280     0463  2 !                   bits 4-6 indicate summary/status/characteristics
;  281     0464  2 !                 For Function CHANGE
;  282     0465  2 !                   bit 6 indicates whether set/define or clear/purge
;  283     0466  2 !                   bit 7 indicates whether permanent or volatile,
;  284     0467  2 !                         accept only volatile
;  285     0468  2 !   Bytes 3-17  Module name ASCIC string, "CONFIGURATOR"
;  286     0469  2 !   Bytes 18,19 Code for circuit
;  287     0470  2 !   Byte 20     Code for Known, or count for circuit name
;  288     0471  2 !   Bytes 21-22 or Next two bytes after circuit name:
;  289     0472  2 !               code for surveillance
;  290     0473  2 !   Next byte   surveillance code, 0-Enabled, 1-Disabled
;  291     0474  2 !
;  292     0475  2
;  293     0476  2 !
;  294     0477  2 !   Check the specified option and accept only SET or SHOW
;  295     0478  2 !
;  296     0479  2     FUNCTION = .NICE [0,0,8,0];
;  297     0480  2     OPTION = .NICE [0,8,8,0];
;  298     0481  2
;  299     0482  2     IF .OPTION [NMA$V_OPT_PER]                  ! There is no permanent data base so
;  300     0483  2     THEN                                        ! DEFINE, LIST or PURGE not permitted
;  301     0484  3         BEGIN
;  302     0485  3         CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_FUN, 0, 0, 0);
;  303     0486  3         RETURN TRUE;
;  304     0487  2         END;
;  305     0488  2
;  306     0489  2     IF .FUNCTION EQL NMA$C_FNC_CHA              ! If function is CHANGE, accept only SET
;  307     0490  2     THEN
;  308     0491  3         BEGIN
;  309     0492  3         IF .OPTION [NMA$V_OPT_CLE]
;  310     0493  3         THEN
;  311     0494  4             BEGIN                               ! CLEAR not permitted
;  312     0495  4             CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_FUN, 0, 0, 0);
;  313     0496  4             RETURN TRUE;
;  314     0497  3             END;
;  315     0498  3
;  316     0499  3         PROCESSING_SHOW = FALSE;                ! Must be a SET
;  317     0500  3         END
;  318     0501  2     ELSE
;  319     0502  3         BEGIN
;  320     0503  3         IF .FUNCTION EQL NMA$C_FNC_REA
;  321     0504  3         THEN
;  322     0505  4             BEGIN
;  323     0506  4             PROCESSING_SHOW = TRUE;             ! It's a SHOW
;  324     0507  4             SHOW_INFO = .OPTION [NMA$V_OPT_INF];  ! Characteristics, Summary or Status
;  325     0508  4             END
;  326     0509  3         ELSE                                    ! Only accept SET or SHOW
;  327     0510  4             BEGIN
;  328     0511  4             CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_FUN, 0, 0, 0);
;  329     0512  4             RETURN TRUE;
;  330     0513  3             END;
;  331     0514  2         END;
;  332     0515  2
;  333     0516  2 !
```

```
;  334     0517  2 !   Ensure that MODULE CONFIGURATOR was specified
;  335     0518  2 !
;  336     0519  2       IF .OPTION [NMA$V_OPT_ENT] NEQ NMA$C_ENT_MOD
;  337     0520  2       THEN
;  338     0521  2           BEGIN
;  339     0522  2           CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_FUN, 0, 0, 0);
;  340     0523  2           RETURN TRUE;
;  341     0524  2           END;
;  342     0525  2
;  343     0526  2       IF NOT CH$EQL (.NICE [0,16,8,0], NICE [0,24,8,0], .CONF [0], CONF [1])
;  344     0527  2       THEN
;  345     0528  2           BEGIN
;  346     0529  2           CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_FUN, 0, 0, 0);
;  347     0530  2           RETURN TRUE;
;  348     0531  2           END;
;  349     0532  2
;  350     0533  2 !
;  351     0534  2 !   Check for CIRCUIT Circuit-name, or for KNOWN CIRCUITS
;  352     0535  2 !
;  353     0536  2
;  354     0537  2       If .NICE [15,0,16,0] NEQ NMA$C_PCCN_CIR
;  355     0538  2       THEN
;  356     0539  2           BEGIN
;  357     0540  2           CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_IDE, NMA$C_ENT_CIR, 0, 0);
;  358     0541  2           RETURN TRUE;
;  359     0542  2           END;
;  360     0543  2       IF .NICE [16,8,8,1] EQL NMA$C_ENT_KNO
;  361     0544  2       THEN                                        ! Known circuits
;  362     0545  2           BEGIN
;  363     0546  2           KNOWN = TRUE;
;  364     0547  2           NICE_SURVEILLANCE = NICE [16,16,8,0];
;  365     0548  2           END
;  366     0549  2       ELSE                                        ! Parse and store ASCIC circuit name
;  367     0550  2           BEGIN
;  368     0551  2           LOCAL
;  369     0552  2               CIRNAM_LEN,                         ! Use temp store, so that if CNF$GET_ZVM returns a failure,
;  370     0553  2                                                   !  calling routine won't erroneously attempt to deallocate
;  371     0554  2               CIRCUIT_PTR;
;  372     0555  2
;  373     0556  2           KNOWN = FALSE;
;  374     0557  2           CIRNAM_LEN = .NICE [16,8,8,0];
;  375   P 0558  2           EXECUTE (
;  376     0559  2             CNF$GET_ZVM ( CIRNAM_LEN, CIRNAM_DSC [DSC$A_POINTER]) );
;  377     0560  2           CIRNAM_DSC [DSC$W_LENGTH] = .CIRNAM_LEN;
;  378     0561  2           CIRCUIT_PTR = NICE [16,16,8,0];
;  379     0562  2
;  380     0563  2           !
;  381     0564  2           !   Check the length of the circuit name and ensure that it does
;  382     0565  2           !   not extend past the end of the NICE message.
;  383     0566  2           !
;  384     0567  2           IF  (.CIRCUIT_PTR - .NICE)              ! Address of circuit minus start of NICE gives length of NIC
;  385     0568  2           + .CIRNAM_DSC [DSC$W_LENGTH]            ! plus length of circuit name gives length of NICE message u
;  386     0569  2               GTR .IRB [IRB$W_IOSB1]              ! Does circuit name extend off end of NICE message?
;  387     0570  2           THEN
;  388     0571  2               BEGIN
;  389     0572  2               CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_IDE, NMA$C_ENT_CIR, 0, 0);
;  390     0573  2               RETURN TRUE;
```

```
;  391     0574  3              END;
;  392     0575  3
;  393     0576  3          CH$MOVE ( .CIRNAM_DSC [DSC$W_LENGTH], .CIRCUIT_PTR, .CIRNAM_DSC [DSC$A_POINTER]);
;  394     0577  3
;  395     0578  3
;  396     0579  3          !
;  397     0580  3          !   Surveillance code and value follows after circuit name
;  398     0581  3          !
;  399     0582  3          NICE_SURVEILLANCE = .CIRCUIT_PTR + . CIRNAM_DSC [DSC$W_LENGTH];
;  400     0583  3          END;
;  401     0584  2
;  402     0585  2      !
;  403     0586  2      !   Compute length of remaining unparsed NICE message.
;  404     0587  2      !
;  405     0588  2      LEN_REMAINING = .IRB [IRB$W_IOSB1] - (.NICE_SURVEILLANCE - .NICE);
;  406     0589  2
;  407     0590  2  !
;  408     0591  2  !   If SHOW then check that nothing is left unprocessed
;  409     0592  2  !
;  410     0593  2      IF .PROCESSING_SHOW
;  411     0594  2      THEN
;  412     0595  3          BEGIN
;  413     0596  3          IF .LEN_REMAINING NEQ 0                          ! There is excess NICE message left unprocessed for
;  414     0597  3          THEN
;  415     0598  4              BEGIN
;  416     0599  4              CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_SIZ, 0, 0, 0);
;  417     0600  4              RETURN TRUE;
;  418     0601  4              END
;  419     0602  3          ELSE
;  420     0603  3              EXECUTE (CNF$PROCESS_SHOW (.IRB, .KNOWN, .CIRNAM_DSC, .SHOW_INFO));
;  421     0604  3          END
;  422     0605  2      ELSE
;  423     0606  2
;  424     0607  2      !
;  425     0608  2      !   For SET, check for SURVEILLANCE TYPE (enabled = 0, disabled = 1)
;  426     0609  2      !   and dispatch to either enable or disable surveillance.
;  427     0610  2      !
;  428     0611  3          BEGIN
;  429     0612  3          IF .LEN_REMAINING NEQ 0                          ! If SURVEILLANCE not given, default to ENABLE
;  430     0613  3          THEN
;  431     0614  4              BEGIN
;  432     0615  4              IF .LEN_REMAINING NEQ 3
;  433     0616  4              THEN                                         ! Must be a word for surveillance code and
;  434     0617  5                  BEGIN                                    ! a byte for the surveillance value.
;  435     0618  5                  CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_PMS, NMA$C_PCCN_SUR, 0, 0);
;  436     0619  5                  RETURN TRUE;
;  437     0620  4                  END;
;  438     0621  4              IF .NICE_SURVEILLANCE [0,0,16,0] NEQ NMA$C_PCCN_SUR
;  439     0622  4              THEN
;  440     0623  5                  BEGIN
;  441     0624  5                  CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_PMS, NMA$C_PCCN_SUR, 0, 0);
;  442     0625  5                  RETURN TRUE;
;  443     0626  4                  END;
;  444     0627  4
;  445     0628  4              IF .NICE_SURVEILLANCE [0,16,8,0] EQL NMA$C_SUR_ENA
;  446     0629  4              THEN
;  447     0630  5                  EXECUTE (CNF_ENABLE_SURVEILLANCE (.IRB, .KNOWN, .CIRNAM_DSC))
```

```
;  448       0631  4            ELSE
;  449       0632  4                EXECUTE (CNF_DISABLE_SURVEILLANCE (.IRB, .KNOWN, .CIRNAM_DSC));
;  450       0633  4            END
;  451       0634  3        ELSE                                        ! Default to setting surveillance enabled
;  452       0635  3            EXECUTE (CNF_ENABLE_SURVEILLANCE (.IRB, .KNOWN, .CIRNAM_DSC));
;  453       0636  2        END;
;  454       0637  2
;  455       0638  2    RETURN TRUE;
;  456       0639  1    END;                                            ! Routine process_request


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

00 00 52 4F 54 41 52 55 47 49 46 4E 4F 43 0C 00004 P.AAB:  .ASCII  <12>\CONFIGURATOR\<0><0><0>          ;
                                          00 00013                                                   ;
                           45 43 41 52 54 00014 P.AAD:  .ASCII  \TRACE\                             ;
                                             00019        .BLKB   3                                  ;
                                  00000005  0001C P.AAC:  .LONG   5                                  ;
                                  00000000' 00020        .ADDRESS P.AAD                              ;
74 73 65 75 71 65 72 5F 73 73 65 63 6F 72 70 00024 P.AAF:  .ASCII  \process_request\                 ;
                                             00033        .BLKB   1                                  ;
                                  0000000F  00034 P.AAE:  .LONG   15                                 ;
                                  00000000' 00038        .ADDRESS P.AAF                              ;

                                                  CONF=             P.AAB


                                                        .PSECT  $CODE$,NOWRT,2

                                     0FFC 00000 PROCESS_REQUEST:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11   ; 0404
                   5E            08 C2 00002            SUBL2   #8, SP                                 ;
                        0000'    CF 9F 00005            PUSHAB  P.AAE                                  ; 0444
                        0000'    CF 9F 00009            PUSHAB  P.AAC                                  ; 0443
                                 01 DD 0000D            PUSHL   #1                                     ;
             0000G   CF          03 FB 0000F            CALLS   #3, CNF$TRACE                          ;
                     58      04  AC D0 00014            MOVL    IRB, R8                                ; 0446
                     56      65  A8 9E 00018            MOVAB   101(R8), NICE                          ;
                     12      0E  A8 B1 0001C            CMPW    14(R8), #18                            ; 0449
                                 09 18 00020            BGEQ    1$                                     ;
                                 7E 7C 00022            CLRQ    -(SP)                                  ; 0452
                                 7E D4 00024            CLRL    -(SP)                                  ;
                     7E          02 CE 00026            MNEGL   #2, -(SP)                              ;
                                 45 11 00029            BRB     5$                                     ;
                     53          66 90 0002B 1$:        MOVB    (NICE), FUNCTION                       ; 0479
                     52      01  A6 90 0002E            MOVB    1(NICE), OPTION                        ; 0480
                                 35 19 00032            BLSS    4$                                     ; 0482
                     13          53 91 00034            CMPB    FUNCTION, #19                          ; 0489
                                 08 12 00037            BNEQ    2$                                     ;
          2C         52          06 E0 00039            BBS     #6, OPTION, 4$                         ; 0492
                                 5B D4 0003D            CLRL    PROCESSING_SHOW                        ; 0499
                                 0D 11 0003F            BRB     3$                                     ; 0489
                     14          53 91 00041 2$:        CMPB    FUNCTION, #20                          ; 0503
                                 23 12 00044            BNEQ    4$                                     ;
                     5B          01 D0 00046            MOVL    #1, PROCESSING_SHOW                    ; 0506
    6E        52     03          04 EF 00049            EXTZV   #4, #3, OPTION, SHOW_INFO              ; 0507
```

```
                                                00  ED  0004E 3$:     CMPZV   #0, #3, OPTION, #4                      ; 0519
                                                14  12  00053       BNEQ    4$
                                                51  02  A6  9A 00055       MOVZBL  2(NICE), R1                     ; 0526
                                                50  0000' CF  9A 00059       MOVZBL  CONF, R0
                                            50  00  03  A6  51  2D 0005E       CMPC5   R1, 3(NICE), #0, R0, CONF+1
                                                    0000' CF  00064
                                                09  13  00067       BEQL    6$
                                                7E  7C  00069 4$:     CLRQ    -(SP)                               ; 0529
                                                7E  D4  0006B       CLRL    -(SP)
                                                7E  01  CE  0006D       MNEGL   #1, -(SP)
                                                7C  11  00070 5$:     BRB     12$
                                        0064    8F  0F  A6  B1  00072 6$:     CMPW    15(NICE), #100                  ; 0537
                                                48  12  00078       BNEQ    9$
                                                53  12  A6  9E 0007A       MOVAB   18(R6), R3                      ; 0547
                                          FF    8F  11  A6  91  0007E       CMPB    17(NICE), #-1                   ; 0543
                                                08  12  00083       BNEQ    7$
                                                59  01  D0  00085       MOVL    #1, KNOWN                       ; 0546
                                                5A  53  D0  00088       MOVL    R3, NICE_SURVEILLANCE           ; 0547
                                                4B  11  0008B       BRB     11$                                 ; 0543
                                                59  D4  0008D 7$:     CLRL    KNOWN                               ; 0556
                                            04  AE  11  A6  9A 0008F       MOVZBL  17(NICE), CIRNAM_LEN            ; 0557
                                                52  08  AC  D0  00094       MOVL    CIRNAM_DSC, R2                  ; 0559
                                                    04  A2  9F  00098       PUSHAB  4(R2)
                                                    08  AE  9F  0009B       PUSHAB  CIRNAM_LEN
                                        0000G   CF  02  FB  0009E       CALLS   #2, CNF$GET_ZVM
                                                01  50  E8  000A3       BLBS    STATUS, 8$
                                                    04  000A6       RET
                                            08  BC  04  AE  B0  000A7 8$:     MOVW    CIRNAM_LEN, @CIRNAM_DSC         ; 0560
                                                57  53  D0  000AC       MOVL    R3, CIRCUIT_PTR                 ; 0561
                                        50      57  56  C3  000AF       SUBL3   NICE, CIRCUIT_PTR, R0           ; 0567
                                                51  08  BC  3C  000B3       MOVZWL  @CIRNAM_DSC, R1                 ; 0568
                                                50  51  C0  000B7       ADDL2   R1, R0
                                    50  0E  AB      10  00  EC  000BA       CMPV    #0, #16, 14(R8), R0             ; 0569
                                                09  18  000C0       BGEQ    10$
                                                7E  7C  000C2 9$:     CLRQ    -(SP)                               ; 0572
                                                03  DD  000C4       PUSHL   #3
                                                7E  09  CE  000C6       MNEGL   #9, -(SP)
                                                4B  11  000C9       BRB     16$
                                04  B2          67  08  BC  28  000CB 10$:    MOVC3   @CIRNAM_DSC, (CIRCUIT_PTR), @4(R2)  ; 0576
                                                5A  08  BC  3C  000D1       MOVZWL  @CIRNAM_DSC, NICE_SURVEILLANCE  ; 0582
                                                5A  57  C0  000D5       ADDL2   CIRCUIT_PTR, NICE_SURVEILLANCE
                                                56  5A  C2  000D8 11$:     SUBL2   NICE_SURVEILLANCE, R6           ; 0588
                                                50  0E  A8  32  000DB       CVTWL   14(R8), LEN_REMAINING
                                                50  56  C0  000DF       ADDL2   R6, LEN_REMAINING
                                                1A  5B  E9  000E2       BLBC    PROCESSING_SHOW, 14$            ; 0593
                                                    09  13  000E5       BEQL    13$                             ; 0596
                                                    7E  7C  000E7       CLRQ    -(SP)                           ; 0599
                                                    7E  D4  000E9       CLRL    -(SP)
                                                7E      04  CE  000EB       MNEGL   #4, -(SP)
                                                    26  11  000EE 12$:     BRB     16$
                                                    6E  DD  000F0 13$:     PUSHL   SHOW_INFO                       ; 0603
                                                08      AC  DD  000F2       PUSHL   CIRNAM_DSC
                                                7E      58  7D  000F5       MOVQ    R8, -(SP)
                                        0000G   CF      04  FB  000F8       CALLS   #4, CNF$PROCESS_SHOW
                                                    3D  11  000FD       BRB     19$
                                                    30  13  000FF 14$:     BEQL    18$                             ; 0612
                                                03      50  D1  00101       CMPL    LEN_REMAINING, #3               ; 0615
```

```
                                   07  12 00104          BNEQ    15$                                 ;
               006E  8F            6A  B1 00106          CMPW    (NICE_SURVEILLANCE), #110           ; 0621
                                   12  13 0010B          BEQL    17$                                 ;
                                   7E  7C 0010D  15$:    CLRQ    -(SP)                               ; 0624
                     7E        6E  8F  9A 0010F          MOVZBL  #110, -(SP)                         ;
                     7E            1D  CE 00113          MNEGL   #29, -(SP)                          ;
                                   58  DD 00116  16$:    PUSHL   R8                                  ;
               0000G CF            05  FB 00118          CALLS   #5, CNF$BUFR_ERR_MSG                ;
                                   20  11 0011D          BRB     20$                                 ; 0625
                               02  AA  95 0011F  17$:    TSTB    2(NICE_SURVEILLANCE)                ; 0628
                                   0D  13 00122          BEQL    18$                                 ;
                               08  AC  DD 00124          PUSHL   CIRNAM_DSC                          ; 0632
                     7E            58  7D 00127          MOVQ    R8, -(SP)                           ;
               0000V CF            03  FB 0012A          CALLS   #3, CNF_DISABLE_SURVEILLANCE        ;
                                   0B  11 0012F          BRB     19$                                 ;
                               08  AC  DD 00131  18$:    PUSHL   CIRNAM_DSC                          ; 0635
                     7E            58  7D 00134          MOVQ    R8, -(SP)                           ;
               0000V CF            03  FB 00137          CALLS   #3, CNF_ENABLE_SURVEILLANCE         ;
                     03            50  E9 0013C  19$:    BLBC    STATUS, 21$                         ;
                     50            01  D0 0013F  20$:    MOVL    #1, R0                              ; 0638
                                       04 00142  21$:    RET                                         ; 0639

; Routine Size:  323 bytes,    Routine Base:  $CODE$ + 0041
```

```
;  458     0640  1 %SBTTL  'cnf_enable_surveillance  '
;  459     0641  1 ROUTINE CNF_ENABLE_SURVEILLANCE (IRB, KNOWN, CIRCUITNAM_DSC) =
;  460     0642  1
;  461     0643  1 !++
;  462     0644  1 !
;  463     0645  1 !   Jacket routine to ensure common error recovery and memory
;  464     0646  1 !   deallocation for the enabling of surveillance logic.
;  465     0647  1 !
;  466     0648  1 !       irb             Interrupt request block, containing request context
;  467     0649  1 !
;  468     0650  1 !       known           If true, then set surveillance for all circuits
;  469     0651  1 !
;  470     0652  1 !       circuitnam_dsc  Descriptor for name of circuit to set surveillance on.
;  471     0653  1 !
;  472     0654  1 !   Always return success, any errors are buffered and then sent to
;  473     0655  1 !   connectee.
;  474     0656  1 !--
;  475     0657  1
;  476     0658  2     BEGIN
;  477     0659  2     LOCAL
;  478     0660  2         CIRCUIT : REF BBLOCK,
;  479     0661  2         STATUS;
;  480     0662  2     MAP
;  481     0663  2         CIRCUITNAM_DSC : REF BBLOCK;
;  482     0664  2
;  483     0665  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;  484     0666  2         $DESCRIPTOR ('cnf_enable_surveillance'));
;  485     0667  2
;  486     0668  2     STATUS = ENABLE_SURVEILLANCE (.IRB, .KNOWN, .CIRCUITNAM_DSC);
;  487     0669  2     IF NOT .STATUS
;  488     0670  2     THEN                                ! buffer up an error response
;  489     0671  2         CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_MPR, 0, .STATUS)
;  490     0672  2     ELSE                                ! Buffer up the 'Success' NICE response
;  491     0673  2         CNF$BUFR_NICE_MSG (.IRB, SUCCESS_NICE_DSC, 0);
;  492     0674  2
;  493     0675  2     !
;  494     0676  2     !   Check to ensure that there is still something under surveillance,
;  495     0677  2     !   otherwise, clear flag so that when execution returns to primary loop
;  496     0678  2     !   in CNFMAIN it will terminate.
;  497     0679  2     !
;  498     0680  2     CNF$B_SURVEILLANCE_SET = FALSE;     ! Assume none has been set
;  499     0681  2     CIRCUIT = .CNF$GQ_CIRSURLST;        ! first circuit in list
;  500     0682  2     WHILE .CIRCUIT NEQ CNF$GQ_CIRSURLST DO              ! For every circuit
;  501     0683  3         BEGIN
;  502     0684  3         IF .CIRCUIT [CIR$B_SURVEIL] EQL NMA$C_SUR_ENA   ! If surveillance is enabled
;  503     0685  3         THEN CNF$B_SURVEILLANCE_SET = TRUE;             ! Then ensure that image execution will continue
;  504     0686  3         CIRCUIT = .CIRCUIT [CIR$L_LINK];                ! Next circuit in list
;  505     0687  2         END;                            ! WHILE traversing Circuit linked list
;  506     0688  2
;  507     0689  2     RETURN TRUE;
;  508     0690  1     END;                                ! Routine cnf_enable_surveillance


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2                                    ;

                    45  43  41  52  54  0003C P.AAH:  .ASCII  \TRACE\                                            ;
```

```
                                                00041                   .BLKB   3
                                        00000005  00044 P.AAG:          .LONG   5
                                        00000000' 00048                 .ADDRESS P.AAH                                  ;
76 72 75 73 5F 65 6C 62 61 6E 65 5F 66 6E 63    0004C P.AAJ:          .ASCII  \cnf_enable_surveillance\                       ;
                     65 63 6E 61 6C 6C 69 65    00058                                                                   ;
                                                00063                   .BLKB   1                                       ;
                                        00000017  00064 P.AAI:          .LONG   23                                      ;
                                        00000000' 00068                 .ADDRESS P.AAJ                                  ;


                                                                        .PSECT  $CODE$,NOWRT,2

                                                0000 00000 CNF_ENABLE_SURVEILLANCE:
                                                                        .WORD   Save nothing                            ; 0641
                                        0000'   CF  9F 00002            PUSHAB  P.AAI                                   ; 0666
                                        0000'   CF  9F 00006            PUSHAB  P.AAG                                   ; 0665
                                                01  DD 0000A            PUSHL   #1                                      ;
                                    0000G   CF      03  FB 0000C        CALLS   #3, CNF$TRACE                           ;
                                            7E      08  AC 7D 00011    MOVQ    KNOWN, -(SP)                            ; 0668
                                                04  AC DD 00015        PUSHL   IRB                                     ;
                                    0000V   CF      03  FB 00018        CALLS   #3, ENABLE_SURVEILLANCE                 ;
                                            11      50  E8 0001D        BLBS    STATUS, 1$                              ; 0669
                                                    50  DD 00020        PUSHL   STATUS                                  ; 0671
                                                    7E  D4 00022        CLRL    -(SP)                                   ;
                                            7E      05  CE 00024        MNEGL   #5, -(SP)                               ;
                                                04  AC DD 00027        PUSHL   IRB                                     ;
                                    0000G   CF      04  FB 0002A        CALLS   #4, CNF$BUFR_ERR_MSG                    ;
                                                    0E  11 0002F        BRB     2$                                      ;
                                                    7E  D4 00031 1$:    CLRL    -(SP)                                   ; 0673
                                        0000'   CF  9F 00033            PUSHAB  SUCCESS_NICE_DSC                        ;
                                          04    AC DD 00037             PUSHL   IRB                                     ;
                                    0000G   CF      03  FB 0003A        CALLS   #3, CNF$BUFR_NICE_MSG                   ;
                                        0000G   CF  D4 0003F 2$:        CLRL    CNF$B_SURVEILLANCE_SET                  ; 0680
                                    51  0000G   CF  D0 00043            MOVL    CNF$GQ_CIRSURLST, CIRCUIT               ; 0681
                                    50  0000G   CF  9E 00048 3$:        MOVAB   CNF$GQ_CIRSURLST, R0                    ; 0682
                                    50          51  D1 0004D            CMPL    CIRCUIT, R0                             ;
                                                0F  13 00050            BEQL    5$                                      ;
                                          0A    A1  95 00052            TSTB    10(CIRCUIT)                             ; 0684
                                                05  12 00055            BNEQ    4$                                      ;
                                    0000G   CF      01  D0 00057        MOVL    #1, CNF$B_SURVEILLANCE_SET              ; 0685
                                    51              61  D0 0005C 4$:    MOVL    (CIRCUIT), CIRCUIT                      ; 0686
                                                    E7  11 0005F        BRB     3$                                      ; 0682
                                    50              01  D0 00061 5$:    MOVL    #1, R0                                  ; 0689
                                                        04 00064        RET                                             ; 0690

; Routine Size:  101 bytes,    Routine Base:  $CODE$ + 0184
```

```
;  510          0691  1 %SBTTL  'enable_surveillance  '
;  511          0692  1 ROUTINE ENABLE_SURVEILLANCE (IRB, KNOWN, CIRCUITNAM_DSC) =
;  512          0693  1
;  513          0694  1 !++
;  514          0695  1 !
;  515          0696  1 !  Perform some checking before calling the routine which will
;  516          0697  1 !  handle the actual establishing of surveillance on a circuit by
;  517          0698  1 !  first determining if the requested circuit is an NI circuit.
;  518          0699  1 !  If known was specified, then discover all the NI circuits available.
;  519          0700  1 !
;  520          0701  1 !       irb             Interrupt request block, containing request context
;  521          0702  1 !
;  522          0703  1 !       known           If true, then set surveillance for all circuits
;  523          0704  1 !
;  524          0705  1 !       circuitnam_dsc  Descriptor for name of circuit to set surveillance on.
;  525          0706  1 !
;  526          0707  1 !--
;  527          0708  2     BEGIN
;  528          0709  2     MAP
;  529          0710  2         CIRCUITNAM_DSC : REF BBLOCK;
;  530          0711  2
;  531          0712  2     MACRO
;  532          0713  2         STRINGS_ARE_EQUAL (COMMAND) = NOT (COMMAND)%;
;  533          0714  2
;  534          0715  2     LITERAL
;  535          0716  2         NFB_ARGS = 4,
;  536          0717  2         NFBSIZ = NFB$C_LENGTH + NFB_ARGS * 4,           ! Network function block size
;  537          0718  2         P2BUFSIZ = 4 + NFB$C_CTX_SIZE,
;  538          0719  2         P4BUFSIZ = 512;
;  539          0720  2
;  540          0721  2     LOCAL
;  541          0722  2         CIRNAM_DSC :    VECTOR [2],
;  542          0723  2         DEVNAM_DSC :    VECTOR [2],
;  543          0724  2         IOSB :          BBLOCK [8],                     ! IO status block
;  544          0725  2         NFB : BBLOCK [NFBSIZ],                          ! Network function block
;  545          0726  2                                                         ! with 4 optional field ID longwords
;  546          0727  2         NFB_DESC :      VECTOR [2],                     ! Descriptor of NFB
;  547          0728  2         PTR,
;  548          0729  2         P2BUF_DSC :     VECTOR [2],                     ! Descriptor of P2 buffer
;  549          0730  2         P2BUF : BBLOCK [P2BUFSIZ],
;  550          0731  2         P4BUF_DSC :     VECTOR [2],                     ! Descriptor of P4 buffer
;  551          0732  2         P4BUF : BBLOCK [P4BUFSIZ],
;  552          0733  2         SEARCHING,
;  553          0734  2         STATUS,
;  554          0735  2         STATE,                                          ! Store circuit state
;  555          0736  2         TYPE;                                           ! Store circuit type
;  556          0737  2
;  557          0738  2     CNFSTRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;  558          0739  2         $DESCRIPTOR ('enable_surveillance'));
;  559          0740  2
;  560          0741  2
;  561          0742  2 !
;  562          0743  2 !   Translate circuit name to physical device name
;  563          0744  2 !
;  564          0745  2     CH$FILL (0, NFBSIZ, NFB);
;  565          0746  2
;  566          0747  2     NFB [NFB$B_FCT]         = NFB$C_FC_SHOW;            ! Set function to SHOW
```

```
;  567          0748  2     NFB [NFB$B_DATABASE]   = NFB$C_DB_CRI;              ! Circuit database
;  568          0749  2     NFB [NFB$B_OPER]       = NFB$C_OP_EQL;              ! Criteria for a match
;  569          0750  2     NFB [NFB$V_MULT]       = TRUE;                      !
;  570          0751  2     NFB [NFB$L_SRCH_KEY]   = NFB$C_WILDCARD;            !
;  571          0752  2 !   NFB [NFB$L_SRCH2_KEY]  = NFB$C_WILDCARD;            !
;  572          0753  2 !   NFB [NFB$B_OPER2]      = NFB$C_OP_EQL;              ! Criteria for a match
;  573          0754  2     NFB [NFB$L_FLDID]      = NFB$C_CRI_TYP;             ! Circuit type
;  574          0755  2     NFB [NFB$L_FLDID] + 4  = NFB$C_CRI_STA;             ! Circuit state
;  575          0756  2     NFB [NFB$L_FLDID] + 8  = NFB$C_CRI_NAM;             ! Circuit name
;  576          0757  2     NFB [NFB$L_FLDID] + 12  = NFB$C_CRI_VMSNAM;         ! Circuit device name
;  577          0758  2
;  578          0759  2
;  579          0760  2     NFB_DESC [0] = NFBSIZ;      ! Set up descriptor for NFB
;  580          0761  2     NFB_DESC [1] = NFB;
;  581          0762  2
;  582          0763  2     P2BUF_DSC [0] = P2BUFSIZ;
;  583          0764  2     P2BUF_DSC [1] = P2BUF;
;  584          0765  2     CH$FILL (0, P2BUFSIZ, P2BUF);
;  585          0766  2     P4BUF_DSC [0] = P4BUFSIZ;
;  586          0767  2     P4BUF_DSC [1] = P4BUF;
;  587          0768  2
;  588          0769  2     SEARCHING = TRUE;                                   ! If searching for specific
;  589          0770  2                                                         !  circuit, keep calling NETACP
;  590          0771  2
;  591          0772  2     !
;  592          0773  2     !   Call the NETACP and get a buffer full of circuit names and
;  593          0774  2     !   corresponding devices.  Keep calling until it returns
;  594          0775  2     !   SS$_ENDOFFILE.
;  595          0776  2     !
;  596          0777  2     WHILE .SEARCHING DO
;  597          0778  2         BEGIN
;  598          0779  2
;  599          0780  2         CH$FILL (0, P4BUFSIZ, P4BUF);
;  600          0781  2
;  601        P 0782  2         STATUS = $QIOW ( FUNC = IO$_ACPCONTROL, ! Obtain circuit name and circuit device name
;  602        P 0783  2                CHAN = .CNF$W_NETCHAN,           ! Use assigned channel
;  603        P 0784  2                EFN = CNF$C_SYNCH_EFN,           ! Synchronous Event flag number
;  604        P 0785  2                IOSB = IOSB,
;  605        P 0786  2                P1 = NFB_DESC,                   ! Network function block
;  606        P 0787  2                P2 = P2BUF_DSC,                  ! Work space
;  607          0788  2                P4 = P4BUF_DSC);                 ! Buffer for return strings
;  608          0789  2
;  609          0790  2         IF .STATUS
;  610          0791  2         THEN                                        ! successful submission
;  611          0792  2             STATUS = .IOSB [0,0,16,0];              ! pick up final status
;  612          0793  2
;  613          0794  2         IF NOT .STATUS
;  614          0795  2         THEN
;  615          0796  3             BEGIN
;  616          0797  4             IF .STATUS EQL SS$_ENDOFFILE
;  617          0798  4             THEN
;  618          0799  5                 BEGIN
;  619          0800  5                 IF NOT .KNOWN
;  620          0801  5                 THEN
;  621          0802  6                     BEGIN       ! We were looking for a specific circuit and didn't find it.
;  622          0803  6                     CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_IDE, NMA$C_ENT_CIR, 0, CIRCUITNAM_DSC);
;  623          0804  6                     RETURN TRUE;
```

```
;  624     0805  5                    END;
;  625     0806  5                SEARCHING = FALSE;      ! That's all she wrote, so quit the loop
;  626     0807  5                EXITLOOP;
;  627     0808  4                END;
;  628     0809  4            SIGNAL (CNF$_LOGIC, 0, .STATUS);    ! Otherwise, there was an error we'd better report
;  629     0810  4            RETURN .STATUS;
;  630     0811  3            END;
;  631     0812  3
;  632     0813  3        PTR = P4BUF;
;  633     0814  3
;  634     0815  3        !
;  635     0816  3        !   Cycle through circuit names returned in P4 buffer and
;  636     0817  3        !   if KNOWN is selection criteria then set surveillance on all NI
;  637     0818  3        !   circuit devices otherwise search for the requested circuit
;  638     0819  3        !   and set surveillance on it if it is an NI circuit.
;  639     0820  3        !
;  640     0821  3        INCR CIRCUITS FROM 1 TO .P2BUF DO
;  641     0822  3
;  642     0823  4            BEGIN
;  643     0824  4            TYPE = .(.PTR) < 0, 32 >;
;  644     0825  4            PTR = .PTR + 4;
;  645     0826  4
;  646     0827  4            STATE = .(.PTR) < 0, 32 >;                  ! Get circuit state
;  647     0828  4            PTR = .PTR + 4;
;  648     0829  4
;  649     0830  4            CIRNAM_DSC [0] = .(.PTR) < 0, 16 >;         ! Length of circuit name
;  650     0831  4            CIRNAM_DSC [1] = (.PTR) < 16, 8 >;          ! Address of start of circuit name
;  651     0832  4
;  652     0833  4            PTR = .PTR + 2 + .CIRNAM_DSC [0];
;  653     0834  4
;  654     0835  4            DEVNAM_DSC [0] = .(.PTR) < 0, 16 >;         ! Length of circuit name
;  655     0836  4            DEVNAM_DSC [1] = (.PTR) < 16, 8 >;          ! Address of start of circuit name
;  656     0837  4
;  657     0838  4            PTR = .PTR + 2 + .DEVNAM_DSC [0];
;  658     0839  4
;  659     0840  4            !
;  660     0841  4            !   Only interested in NI circuits with State ON
;  661     0842  4            !
;  662     0843  4            IF .TYPE EQL NMA$C_CIRTY_NI AND .STATE EQL NMA$C_STATE_ON
;  663     0844  4            THEN
;  664     0845  4                IF .KNOWN
;  665     0846  4                THEN
;  666     0847  5                    BEGIN                               ! Set surveillance on all NI circuits
;  667     0848  5                    EXECUTE (SURVEIL (CIRNAM_DSC, DEVNAM_DSC));
;  668     0849  5                    END
;  669     0850  4                ELSE
;  670     0851  5                    BEGIN                               ! Looking for a specific circuit
;  671     0852  6                    IF STRINGS_ARE_EQUAL (STR$COMPARE (.CIRCUITNAM_DSC, CIRNAM_DSC))
;  672     0853  5                    THEN
;  673     0854  6                        BEGIN
;  674     0855  6                        EXECUTE (SURVEIL (CIRNAM_DSC, DEVNAM_DSC));
;  675     0856  6                        SEARCHING = FALSE;              ! We got it and can quit now
;  676     0857  6                        EXITLOOP;
;  677     0858  5                        END;
;  678     0859  4                    END;
;  679     0860  3            END;                    ! while INCRementing through QIO return buffer
;  680     0861  2        END;                        ! WHILE performing QIOs to NETACP
```

```
;  681          0862  2
;  682          0863  2     RETURN TRUE;
;  683          0864  1     END;                          ! Routine enable_surveillance


                                                         .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    45  43  41  52  54  0006C P.AAL:  .ASCII  \TRACE\                          ;
                                                        00071        .BLKB   3
                                          00000005      00074 P.AAK:  .LONG   5                               ;
                                          00000000'     00078        .ADDRESS P.AAL                         ;
6C  6C  69  65  76  72  75  73  5F  65  6C  62  61  6E  65  0007C P.AAN:  .ASCII  \enable_surveillance\       ;
                                                    65  63  6E  61   0008B                                    ;
                                                        0008F        .BLKB   1                              ;
                                          00000013      00090 P.AAM:  .LONG   19                              ;
                                          00000000'     00094        .ADDRESS P.AAN                         ;

                                                         .EXTRN  SYS$QIOW

                                                         .PSECT  $CODE$,NOWRT,2

                                        07FC 00000 ENABLE_SURVEILLANCE:
                                                         .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10           ; 0692
                                    5E      FD6C    CE   9E 00002        MOVAB   -660(SP), SP               ;
                                            0000'   CF   9F 00007        PUSHAB  P.AAM                      ; 0739
                                            0000'   CF   9F 0000B        PUSHAB  P.AAK                      ; 0738
                                                    01   DD 0000F        PUSHL   #1                         ;
                                0000G       CF           03   FB 00011  CALLS   #3, CNF$TRACE              ;
            20                      00          6E       00   2C 00016  MOVC5   #0, (SP), #0, #32, NFB      ; 0745
                                                    C8   AD   0001B                                          ;
                                    C8      AD           22   90 0001D  MOVB    #34, NFB                   ; 0747
                                    CA      AD           04   B0 00021  MOVW    #4, NFB+2                  ; 0748
                                    C9      AD           02   88 00025  BISB2   #2, NFB+1                  ; 0750
                                    CC      AD           01   D0 00029  MOVL    #1, NFB+4                  ; 0751
                                    D8      AD  04010020 8F   D0 0002D  MOVL    #67174432, NFB+16          ; 0754
                                    DC      AD  04010013 8F   D0 00035  MOVL    #67174419, NFB+20          ; 0755
                                    E0      AD  04020041 8F   D0 0003D  MOVL    #67240001, NFB+24          ; 0756
                                    E4      AD  04020042 8F   D0 00045  MOVL    #67240002, NFB+28          ; 0757
                                    C0      AD           20   D0 0004D  MOVL    #32, NFB_DESC              ; 0760
                                    C4      AD       C8  AD   9E 00051  MOVAB   NFB, NFB_DESC+4            ; 0761
                                    B8      AD       44  8F   9A 00056  MOVZBL  #68, P2BUF_DSC             ; 0763
                                    BC      AD     FF74  CD   9E 0005B  MOVAB   P2BUF, P2BUF_DSC+4          ; 0764
            0044    8F                  00          6E       00   2C 00061  MOVC5   #0, (SP), #0, #68, P2BUF  ; 0765
                                                  FF74  CD   00068                                         ;
                                FF6C    CD      0200     8F   3C 0006B  MOVZWL  #512, P4BUF_DSC            ; 0766
                                FF70    CD               6E   9E 00072  MOVAB   P4BUF, P4BUF_DSC+4         ; 0767
                                        58               01   D0 00077  MOVL    #1, SEARCHING              ; 0769
                                        5E               58   E9 0007A 1$:  BLBC   SEARCHING, 4$           ; 0777
            0200    8F                  00          6E       00   2C 0007D  MOVC5   #0, (SP), #0, #512, P4BUF ; 0780
                                                            6E   00084                                       ;
                                                            7E   7C 00085  CLRQ    -(SP)                      ; 0788
                                                  FF6C      CD   9F 00087  PUSHAB  P4BUF_DSC                 ;
                                                            7E   D4 0008B  CLRL    -(SP)                     ;
                                                       B8   AD   9F 0008D  PUSHAB  P2BUF_DSC                ;
                                                       C0   AD   9F 00090  PUSHAB  NFB_DESC                 ;
                                                            7E   7C 00093  CLRQ    -(SP)                     ;
```

```
                                E8  AD  9F 00095          PUSHAB  IOSB                                    ;
                                    38  DD 00098          PUSHL   #56                                     ;
                             7E      0000G  CF  3C 0009A          MOVZWL  CNF$W_NETCHAN, -(SP)            ;
                                 00000000G  8F  DD 0009F          PUSHL   #CNF$C_SYNCH_EFN                ;
                   00000000G  00            0C  FB 000A5          CALLS   #12, SYS$QIOW                   ;
                              57            50  D0 000AC          MOVL    R0, STATUS                      ;
                              07            57  E9 000AF          BLBC    STATUS, 2$                      ; 0790
                              57      E8    AD  3C 000B2          MOVZWL  IOSB, STATUS                    ; 0792
                              3A            57  E8 000B6          BLBS    STATUS, 6$                      ; 0794
                   00000870   8F            57  D1 000B9 2$:      CMPL    STATUS, #2160                   ; 0797
                              1C            12  12 000C0          BNEQ    5$                              ;
                              13      08    AC  E8 000C2          BLBS    KNOWN, 3$                       ; 0800
                              0C    0C      AC  9F 000C6          PUSHAB  CIRCUITNAM_DSC                  ; 0803
                              7E            03  7D 000C9          MOVQ    #3, -(SP)                       ;
                              7E            09  CE 000CC          MNEGL   #9, -(SP)                       ;
                                    04      AC  DD 000CF          PUSHL   IRB                             ;
                        0000G  CF           05  FB 000D2          CALLS   #5, CNF$BUFR_ERR_MSG            ;
                                            02  11 000D7          BRB     4$                              ; 0804
                                            58  D4 000D9 3$:      CLRL    SEARCHING                       ; 0806
                                        008A    31 000DB 4$:      BRW     11$                             ; 0799
                                            57  DD 000DE 5$:      PUSHL   STATUS                          ; 0809
                                            7E  D4 000E0          CLRL    -(SP)                           ;
                                 00000000G  8F  DD 000E2          PUSHL   #CNF$_LOGIC                     ;
                   00000000G  00            03  FB 000E8          CALLS   #3, LIB$SIGNAL                  ;
                              50            57  D0 000EF          MOVL    STATUS, R0                      ; 0810
                                            04  000F2          RET                                     ;
                              56            6E  9E 000F3 6$:      MOVAB   P4BUF, PTR                      ; 0813
                                            52  D4 000F6          CLRL    CIRCUITS                        ; 0821
                                            65  11 000F8          BRB     9$                              ;
                              59            86  7D 000FA 7$:      MOVQ    (PTR)+, TYPE                    ; 0824
                        F8    AD            66  3C 000FD          MOVZWL  (PTR), CIRNAM_DSC               ; 0830
                        FC    AD      02    A6  9E 00101          MOVAB   2(R6), CIRNAM_DSC+4             ; 0831
                   50         56      F8    AD  C1 00106          ADDL3   CIRNAM_DSC, PTR, R0             ; 0833
                              56      02    A0  9E 0010B          MOVAB   2(R0), PTR                      ;
                        F0    AD            66  3C 0010F          MOVZWL  (PTR), DEVNAM_DSC               ; 0835
                        F4    AD      02    A6  9E 00113          MOVAB   2(R6), DEVNAM_DSC+4             ; 0836
                   50         56      F0    AD  C1 00118          ADDL3   DEVNAM_DSC, PTR, R0             ; 0838
                              56      02    A0  9E 0011D          MOVAB   2(R0), PTR                      ;
                              06            59  D1 00121          CMPL    TYPE, #6                        ; 0843
                                            39  12 00124          BNEQ    9$                              ;
                                            5A  D5 00126          TSTL    STATE                           ;
                                            35  12 00128          BNEQ    9$                              ;
                              0F      08    AC  E9 0012A          BLBC    KNOWN, 8$                       ; 0845
                                    F0      AD  9F 0012E          PUSHAB  DEVNAM_DSC                      ; 0848
                                    F8      AD  9F 00131          PUSHAB  CIRNAM_DSC                      ;
                        0000V  CF           02  FB 00134          CALLS   #2, SURVEIL                     ;
                              23            50  E8 00139          BLBS    STATUS, 9$                      ;
                                            04  0013C          RET                                     ;
                                    F8      AD  9F 0013D 8$:      PUSHAB  CIRNAM_DSC                      ; 0852
                                    0C      AC  DD 00140          PUSHL   CIRCUITNAM_DSC                  ;
                   00000000G  00            02  FB 00143          CALLS   #2, STR$COMPARE                 ;
                              12            50  E8 0014A          BLBS    R0, 9$                          ;
                                    F0      AD  9F 0014D          PUSHAB  DEVNAM_DSC                      ; 0855
                                    F8      AD  9F 00150          PUSHAB  CIRNAM_DSC                      ;
                        0000V  CF           02  FB 00153          CALLS   #2, SURVEIL                     ;
                              10            50  E9 00158          BLBC    STATUS, 12$                     ;
                                            58  D4 0015B          CLRL    SEARCHING                       ; 0856
```

```
                                    06  11 0015D          BRB     10$                              ; 0854
            95          52    FF74  CD  F3 0015F 9$:      AOBLEQ  P2BUF, CIRCUITS, 7$              ; 0821
                                  FF12  31 00165 10$:     BRW     1$                               ; 0777
                        50          01  D0 00168 11$:     MOVL    #1, R0                           ; 0863
                                        04 0016B 12$:     RET                                      ; 0864

; Routine Size:  364 bytes,    Routine Base:  $CODE$ + 01E9
```

```
;   685         0865  1 %SBTTL  'surveil  Begin surveillance of circuit'
;   686         0866  1 ROUTINE SURVEIL (CIRNAM_DSC, DEVNAM_DSC) =
;   687         0867  1
;   688         0868  1 !++
;   689         0869  1 !
;   690         0870  1 !  This is the routine that actually initiates surveillance of a circuit.
;   691         0871  1 !  Place circuit name and device in circuit list and initiate surveillance.
;   692         0872  1 !
;   693         0873  1 !       cirnam_dsc      For checking if this circuit is already in our lis*
;   694         0874  1 !                       of circuits that we know about.
;   695         0875  1 !
;   696         0876  1 !       devnam_dsc      Physical device name corresponding to the circuit
;   697         0877  1 !                       for communicating with the driver.
;   698         0878  1 !
;   699         0879  1 !--
;   700         0880  1
;   701         0881  2     BEGIN
;   702         0882  2     LOCAL
;   703         0883  2         CIR :           REF BBLOCK,
;   704         0884  2         P2_DESC :       BBLOCK [DSC$C_S_BLN],
;   705         0885  2         STATUS;
;   706         0886  2
;   707         0887  2     LITERAL
;   708         0888  2         P2BUFLEN = 72,
;   709         0889  2         REMOTE_CONSOLE_PROTOCOL = %X'260';
;   710         0890  2
;   711         0891  2     OWN
;   712         0892  2         !
;   713         0893  2         !   P2 buffer for talking with the device driver
;   714         0894  2         !
;   715         0895  2         P2BUF : BBLOCK [P2BUFLEN]
;   716         0896  2                 INITIAL (
;   717         0897  2                         WORD (NMA$C_PCLI_BUS), 64,
;   718         0898  2                         WORD (NMA$C_PCLI_BFN), 1,
;   719         0899  2                         WORD (NMA$C_PCLI_PRM), NMA$C_STATE_OFF,
;   720         0900  2                         WORD (NMA$C_PCLI_MLT), NMA$C_STATE_OFF,
;   721         0901  2                         WORD (NMA$C_PCLI_DCH), NMA$C_STATE_OFF,
;   722         0902  2                         WORD (NMA$C_PCLI_CRC), NMA$C_STATE_ON,
;   723         0903  2                         WORD (NMA$C_PCLI_PAD), NMA$C_STATE_ON,
;   724         0904  2                         WORD (NMA$C_PCLI_PTY), REMOTE_CONSOLE_PROTOCOL,
;   725         0905  2                         WORD (NMA$C_PCLI_CON), NMA$C_LINCN_NOR,
;   726         0906  2                         WORD (NMA$C_PCLI_ACC), NMA$C_ACC_SAR,
;   727         0907  2                         WORD (NMA$C_PCLI_MCA),
;   728         0908  2                             WORD (8), WORD (NMA$C_LINMC_SET),
;   729         0909  2                             BYTE (%X'AB'), BYTE (%X'00'),
;   730         0910  2                             BYTE (%X'00'), BYTE (%X'02'),
;   731         0911  2                             BYTE (%X'00'), BYTE (%X'00')
;   732         0912  2                         );
;   733         0913  2
;   734         0914  2
;   735         0915  2
;   736         0916  2     MAP
;   737         0917  2         CIRNAM_DSC :    REF BBLOCK,
;   738         0918  2         DEVNAM_DSC :    REF BBLOCK;
;   739         0919  2
;   740         0920  2
;   741         0921  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
```

```
;  742          0922  2          $DESCRIPTOR ('surveil'));
;  743          0923  2
;  744          0924  2      !
;  745          0925  2      !   Check and see if we already know about this circuit.
;  746          0926  2      !
;  747          0927  2      IF CNF$LOCATE_CIR_BLK (.CIRNAM_DSC, CIR)
;  748          0928  2      THEN
;  749          0929  2          BEGIN                                       ! Its in our list
;  750          0930  2          IF .CIR [CIR$B_SURVEIL] EQL NMA$C_SUR_ENA
;  751          0931  2          THEN RETURN TRUE;                           ! And surveillance is already set
;  752          0932  2          !
;  753          0933  2          !   Else, make sure the buffers were deallocated, since CNF$READ_SYSIDM
;  754          0934  2          !   will report an error if the buffers are there when it goes to
;  755          0935  2          !   allocate new ones.
;  756          0936  2          !   Then skip the circuit block allocation and go to the set up.
;  757          0937  2          !
;  758          0938  2          IF .CIR [CIR$L_SYSIDMBUF] NEQ 0             ! If the buffer there?
;  759          0939  2          THEN
;  760          0940  3              BEGIN                                   ! If this happens, there is probably a timing bug around
;  761          0941  3              CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR'),
;  762          0942  3                  $DESCRIPTOR ('surveil  buffers in place on re-activation'));
;  763          0943  3              EXECUTE (CNF$FREE_VM (%REF(SYSIDM_BUFSIZ), CIR [CIR$L_SYSIDMBUF]) );
;  764          0944  3              EXECUTE (CNF$FREE_VM (%REF(ADRTYP_BUFSIZ), CIR [CIR$L_ADRTYPBUF]) );
;  765          0945  3              END;
;  766          0946  2          END
;  767          0947  2      ELSE
;  768          0948  2          !
;  769          0949  2          !   This is the first we've heard of this circuit, so create a
;  770          0950  2          !   control block for it and fill it in.
;  771          0951  2          !
;  772          0952  2          BEGIN
;  773          0953  2          EXECUTE ( CNF$GET_ZVM ( %REF(CIR$C_LENGTH), CIR));
;  774          0954  2
;  775          0955  2          CIR [CIR$W_SIZE] = CIR$C_LENGTH;
;  776          0956  2
;  777          0957  2          CIR [CIR$W_CIRNAMLEN] = .CIRNAM_DSC [DSC$W_LENGTH];     ! Save the name
;  778          0958  2          CH$MOVE ( .CIRNAM_DSC [DSC$W_LENGTH], .CIRNAM_DSC [DSC$A_POINTER],
;  779          0959  2              CIR [CIR$T_CIRNAM]);
;  780          0960  2          CIR [CIR$W_DEVNAMLEN] = .DEVNAM_DSC [DSC$W_LENGTH];     ! Save the device name
;  781          0961  2          CH$MOVE ( .DEVNAM_DSC [DSC$W_LENGTH], .DEVNAM_DSC [DSC$A_POINTER],
;  782          0962  2              CIR [CIR$T_DEVNAM]);
;  783          0963  2
;  784          0964  2          !
;  785          0965  2          !   Initialize the linked list for holding the system ID messages
;  786          0966  2          !   that will be gathered for this circuit.
;  787          0967  2          !
;  788          0968  2          CIR [CIR$L_SIDFLINK] = CIR [CIR$L_SIDFLINK];
;  789          0969  2          CIR [CIR$L_SIDBLINK] = CIR [CIR$L_SIDFLINK];
;  790          0970  2
;  791          0971  2          !
;  792          0972  2          !   Place in on our list of circuits
;  793          0973  2          !
;  794          0974  2          INSQUE (.CIR, .CNF$GQ_CIRSURLST [1]);
;  795          0975  2          END;
;  796          0976  2
;  797          0977  2      CIR [CIR$B_SURVEIL] = NMA$C_SUR_ENA;            ! Record that surveillance is enabled
;  798          0978  2
```

```
;  799          0979  2      !
;  800          0980  2      !   Assign channel to NI driver
;  801          0981  2      !
;  802          0982  2      STATUS = $ASSIGN (CHAN = CIR [CIR$W_CHAN], DEVNAM = .DEVNAM_DSC);
;  803          0983  2      IF NOT .STATUS
;  804          0984  2      THEN
;  805          0985  3          BEGIN
;  806          0986  3          SIGNAL (CNF$_CHAN, 0, .STATUS);
;  807          0987  3          CIR [CIR$B_SURVEIL] = NMA$C_SUR_DIS;    ! Record that surveillance is disabled
;  808          0988  3          RETURN .STATUS;
;  809          0989  2          END;
;  810          0990  2
;  811          0991  2      !
;  812          0992  2      !   Get ready to talk to device driver
;  813          0993  2      !
;  814          0994  2      P2_DESC = 0;                                ! Zero first longword
;  815          0995  2      P2_DESC [DSC$W_LENGTH] = P2BUFLEN;          ! Set buffer size
;  816          0996  2      P2_DESC [DSC$A_POINTER] = P2BUF;            ! Pointer to buffer
;  817          0997  2
;  818          0998  2      !
;  819          0999  2      !   Issue startup info to driver so that future reads will get only
;  820          1000  2      !   the system ID messages that are broadcast
;  821          1001  2      !
;  822        P 1002  2      STATUS = $QIOW
;  823        P 1003  2          (
;  824        P 1004  2          FUNC = (IO$_SETCHAR OR IO$M_CTRL OR IO$M_STARTUP),
;  825        P 1005  2          CHAN = .CIR [CIR$W_CHAN],
;  826        P 1006  2          EFN = CNF$C_SYNCH_EFN,                  ! Synchronous Event flag number
;  827        P 1007  2          IOSB = CIR [CIR$W_IOSB],
;  828        P 1008  2          P2 = P2_DESC
;  829          1009  2          );
;  830          1010  2
;  831          1011  2      IF NOT .STATUS
;  832          1012  2      THEN
;  833          1013  3          BEGIN
;  834          1014  3          SIGNAL (CNF$_DRVRSTRT, 0, .STATUS);
;  835          1015  3          CIR [CIR$B_SURVEIL] = NMA$C_SUR_DIS;    ! Record that surveillance is disabled
;  836          1016  3          RETURN .STATUS;
;  837          1017  2          END;
;  838          1018  2
;  839          1019  2      STATUS = .CIR [CIR$W_IOSB];                 ! pick up final status
;  840          1020  2      IF NOT .STATUS
;  841          1021  2      THEN
;  842          1022  3          BEGIN
;  843          1023  3          SIGNAL (CNF$_DRVRSTRT, 0, .STATUS);
;  844          1024  3          RETURN .STATUS;
;  845          1025  2          END;
;  846          1026  2
;  847          1027  2
;  848          1028  2      !
;  849          1029  2      !   Record the system time when surveillance began since this
;  850          1030  2      !   info is required on a SHOW request.
;  851          1031  2      !
;  852          1032  2      EXECUTE ($GETTIM (TIMADR = CIR [CIR$Q_ELAPSDTIM]) );
;  853          1033  2
;  854          1034  2      !
;  855          1035  2      !   Issue QIO to device driver to request broadcast messages
```

```
;  856          1036  2     !   and set AST for processing System ID messages as they are read.
;  857          1037  2     !
;  858          1038  2     EXECUTE (CNF$READ_SYSIDM (.CIR));
;  859          1039  2
;  860          1040  2     RETURN TRUE;
;  861          1041  1     END;                                ! Routine surveil


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                  45  43  41  52  54  00098 P.AAP:  .ASCII  \TRACE\                                ;
                                                                      0009D        .BLKB   3
                                                              00000005  000A0 P.AAO:  .LONG   5                                     ;
                                                              00000000' 000A4        .ADDRESS P.AAP                                 ;
                                              6C  69  65  76  72  75  73  000A8 P.AAR:  .ASCII  \surveil\                           ;
                                                                      000AF        .BLKB   1
                                                              00000007  000B0 P.AAQ:  .LONG   7                                     ;
                                                              00000000' 000B4        .ADDRESS P.AAR                                 ;
52  4F  52  52  45  20  2A  2A  2A  20  45  43  41  52  54  000B8 P.AAT:  .ASCII  \TRACE *** ERROR\                                ;
                                                                      000C7        .BLKB   1
                                                              0000000F  000C8 P.AAS:  .LONG   15                                    ;
                                                              00000000' 000CC        .ADDRESS P.AAT                                 ;
72  65  66  66  75  62  20  20  6C  69  65  76  72  75  73  000D0 P.AAV:  .ASCII  \surveil  buffers in place on re-activati\      ;
72  20  6E  6F  20  65  63  61  6C  70  20  6E  69  20  73  000DF                                                                    ;
                        69  74  61  76  69  74  63  61  2D  65  000EE                                                               ;
                                                          6E  6F  000F8        .ASCII  \on\                                        ;
                                                                      000FA        .BLKB   2
                                                              0000002A  000FC P.AAU:  .LONG   42                                    ;
                                                              00000000' 00100        .ADDRESS P.AAV                                 ;

                                                                .PSECT  $OWN$,NOEXE,2

                                                      0AF1  00008 P2BUF:  .WORD   2801                                          ;
                                                  00000040  0000A         .LONG   64
                                                      0451  0000E         .WORD   1105
                                                  00000001  00010         .LONG   1
                                                      0B18  00014         .WORD   2840
                                                  00000001  00016         .LONG   1
                                                      0B19  0001A         .WORD   2841
                                                  00000001  0001C         .LONG   1
                                                      0B1B  00020         .WORD   2843
                                                  00000001  00022         .LONG   1
                                                      0B1C  00026         .WORD   2844
                                                  00000000  00028         .LONG   0
                                                      0B1A  0002C         .WORD   2842
                                                  00000000  0002E         .LONG   0
                                                      0B0E  00032         .WORD   2830
                                                  00000260  00034         .LONG   608
                                                      0456  00038         .WORD   1110
                                                  00000000  0003A         .LONG   0
                                                      0B1E  0003E         .WORD   2846
                                                  00000001  00040         .LONG   1
                                                      0B0F  00044         .WORD   2831
                                                      0008  00046         .WORD   8
                                                      0001  00048         .WORD   1
                                                        AB  0004A         .BYTE   -85
```

```
                                00  0004B          .BYTE   0                                                       ;
                                00  0004C          .BYTE   0                                                       ;
                                02  0004D          .BYTE   2                                                       ;
                                00  0004E          .BYTE   0                                                       ;
                                00  0004F          .BYTE   0                                                       ;

                                                   .EXTRN  SYS$ASSIGN, SYS$GETTIM

                                                   .PSECT  $CODE$,NOWRT,2

                              01FC 00000 SURVEIL:: .WORD   Save R2,R3,R4,R5,R6,R7,R8                               ; 0866
                    58 00000000G   00   9E 00002          MOVAB   LIB$SIGNAL, R8
                    57 00000000G   8F   D0 00009          MOVL    #CNF$_DRVRSTRT, R7
                    5E             10   C2 00010          SUBL2   #16, SP
                             0000'  CF   9F 00013          PUSHAB  P.AAQ                                           ; 0922
                             0000'  CF   9F 00017          PUSHAB  P.AAO                                           ; 0921
                                    01   DD 0001B          PUSHL   #1
                0000G       CF             03   FB 0001D  CALLS   #3, CNF$TRACE
                                     04    AE   9F 00022  PUSHAB  CIR                                             ; 0927
                                    53     04   AC   D0 00025  MOVL    CIRNAM_DSC, R3
                                           53   DD 00029  PUSHL   R3
                0000V       CF             02   FB 0002B  CALLS   #2, CNF$LOCATE_CIR_BLK
                            50             50   E9 00030  BLBC    R0, 3$
                            52       04    AE   D0 00033  MOVL    CIR, R2                                         ; 0930
                                     0A    A2   95 00037  TSTB    10(R2)
                                           03   12 0003A  BNEQ    1$
                                         0122   31 0003C  BRW     12$
                                     38    A2   D5 0003F  1$:     TSTL    56(R2)                                  ; 0938
                                           03   12 00042  BNEQ    2$
                                         0080   31 00044  BRW     6$
                             0000'  CF   9F 00047  2$:     PUSHAB  P.AAU                                           ; 0942
                             0000'  CF   9F 0004B          PUSHAB  P.AAS                                           ; 0941
                                    01   DD 0004F          PUSHL   #1
                0000G       CF             03   FB 00051  CALLS   #3, CNF$TRACE
                                     38    A2   9F 00056  PUSHAB  56(R2)                                          ; 0943
                      04    AE 00000000G   8F   D0 00059  MOVL    #SYSIDM_BUFSIZ, 4(SP)
                                     04    AE   9F 00061  PUSHAB  4(SP)
                0000G       CF             02   FB 00064  CALLS   #2, CNF$FREE_VM
                            27             50   E9 00069  BLBC    STATUS, 4$
                                     3C    A2   9F 0006C  PUSHAB  60(R2)                                          ; 0944
                      04    AE 00000000G   8F   D0 0006F  MOVL    #ADRTYP_BUFSIZ, 4(SP)
                                     04    AE   9F 00077  PUSHAB  4(SP)
                0000G       CF             02   FB 0007A  CALLS   #2, CNF$FREE_VM
                            45             50   E8 0007F  BLBS    STATUS, 6$
                                           04   00082  RET
                                     04    AE   9F 00083  3$:     PUSHAB  CIR                                     ; 0953
                      04    AE       48    8F   9A 00086  MOVZBL  #72, 4(SP)
                                     04    AE   9F 0008B  PUSHAB  4(SP)
                0000G       CF             02   FB 0008E  CALLS   #2, CNF$GET_ZVM
                            01             50   E8 00093  4$:     BLBS    STATUS, 5$
                                           04   00096  RET
                            56       04    AE   D0 00097  5$:     MOVL    CIR, R6                                 ; 0955
                      08    A6       48    8F   9B 0009B  MOVZBW  #72, B(R6)
                      16    A6             63   B0 000A0  MOVW    (R3), 22(R6)                                    ; 0957
                18    A6       04    B3   63   28 000A4  MOVC3   (R3), @4(R3), 24(R6)                           ; 0959
                            50       08    AC   D0 000AA  MOVL    DEVNAM_DSC, R0                                  ; 0960
                      28    A6             60   B0 000AE  MOVW    (R0), 40(R6)
```

```
                2A   A6        04   B0                60  28 000B2          MOVC3   (R0), @4(R0), 42(R6)                ; 0962
                               40   A6        40      60  9E 000B8          MOVAB   64(R6), 64(R6)                      ; 0968
                               44   A6        40      60  9E 000BD          MOVAB   64(R6), 68(R6)                      ; 0969
                          0000G     DF                66  0E 000C2          INSQUE  (R6), @CNF$GQ_CIRSURLST+4           ; 0974
                                    52        04      AE  D0 000C7 6$:      MOVL    CIR, R2                             ; 0977
                                              0A      A2  94 000CB          CLRB    10(R2)                              ;
                                                      7E  7C 000CE          CLRQ    -(SP)                               ; 0982
                                              14      A2  9F 000D0          PUSHAB  20(R2)                              ;
                                              08      AC  DD 000D3          PUSHL   DEVNAM_DSC                          ;
                    00000000G     00                    04  FB 000D6          CALLS   #4, SYS$ASSIGN                      ;
                                  53                    50  D0 000DD          MOVL    R0, STATUS                          ;
                                  0C                    53  E8 000E0          BLBS    STATUS, 7$                          ; 0983
                                                        53  DD 000E3          PUSHL   STATUS                              ; 0986
                                                        7E  D4 000E5          CLRL    -(SP)                               ;
                                        00000000G       8F  DD 000E7          PUSHL   #CNF$_CHAN                          ;
                                                        3E  11 000ED          BRB     8$                                  ;
                                              08        AE  D4 000EF 7$:      CLRL    P2_DESC                             ; 0994
                               08   AE        48      8F  9B 000F2          MOVZBW  #72, P2_DESC                        ; 0995
                               0C   AE      0000'     CF  9E 000F7          MOVAB   P2BUF, P2_DESC+4                    ; 0996
                                                      7E  7C 000FD          CLRQ    -(SP)                               ; 1009
                                                      7E  7C 000FF          CLRQ    -(SP)                               ;
                                              18      AE  9F 00101          PUSHAB  P2_DESC                             ;
                                                      7E  7C 00104          CLRQ    -(SP)                               ;
                                                      7E  D4 00106          CLRL    -(SP)                               ;
                                              0C      A2  9F 00108          PUSHAB  12(R2)                              ;
                                  7E        025A      8F  3C 0010B          MOVZWL  #602, -(SP)                         ;
                                  7E          14      A2  32 00110          CVTWL   20(R2), -(SP)                       ;
                                        00000000G       8F  DD 00114          PUSHL   #CNF$C_SYNCH_EFN                    ;
                    00000000G     00                    0C  FB 0011A          CALLS   #12, SYS$QIOW                       ;
                                  53                    50  D0 00121          MOVL    R0, STATUS                          ;
                                  0F                    53  E8 00124          BLBS    STATUS, 9$                          ; 1011
                                                        53  DD 00127          PUSHL   STATUS                              ; 1014
                                                        7E  D4 00129          CLRL    -(SP)                               ;
                                                        57  DD 0012B          PUSHL   R7                                  ;
                                  68                    03  FB 0012D 8$:      CALLS   #3, LIB$SIGNAL                      ;
                             0A   A2                    01  90 00130          MOVB    #1, 10(R2)                          ; 1015
                                                        10  11 00134          BRB     10$                                 ; 1016
                                  53          0C        A2  32 00136 9$:      CVTWL   12(R2), STATUS                      ; 1019
                                  0D                    53  E8 0013A          BLBS    STATUS, 11$                         ; 1020
                                                        53  DD 0013D          PUSHL   STATUS                              ; 1023
                                                        7E  D4 0013F          CLRL    -(SP)                               ;
                                                        57  DD 00141          PUSHL   R7                                  ;
                                  68                    03  FB 00143          CALLS   #3, LIB$SIGNAL                      ;
                                  50                    53  D0 00146 10$:     MOVL    STATUS, R0                          ; 1024
                                                        04  00149             RET                                         ;
                                              30        A2  9F 0014A 11$:     PUSHAB  48(R2)                              ; 1032
                          00000000G     00                01  FB 0014D          CALLS   #1, SYS$GETTIM                      ;
                                        0D                50  E9 00154          BLBC    STATUS, 13$                         ;
                                                          52  DD 00157          PUSHL   R2                                  ; 1038
                                0000G   CF                01  FB 00159          CALLS   #1, CNF$READ_SYSIDM                 ;
                                        03                50  E9 0015E          BLBC    STATUS, 13$                         ;
                                        50                01  D0 00161 12$:     MOVL    #1, R0                              ; 1040
                                                              04  00164 13$:     RET                                         ; 1041

; Routine Size:  357 bytes,    Routine Base:  $CODE$ + 0355
```

```
;  863          1042  1 %SBTTL  'CNF$LOCATE_CIR_BLK  Locate and return circuit block'
;  864          1043  1 GLOBAL ROUTINE CNF$LOCATE_CIR_BLK (CIRNAMDSC, CIRBLK) =
;  865          1044  1
;  866          1045  1 !++
;  867          1046  1 ! FUNCTIONAL DESCRIPTION:
;  868          1047  1 !
;  869          1048  1 !  Using the descriptor of the ASCII circuit name, search the
;  870          1049  1 !  linked list of circuit blocks to determine the address of
;  871          1050  1 !  the circuit block for the requested circuit name.  If block
;  872          1051  1 !  is not present, return false, else return true.
;  873          1052  1 !
;  874          1053  1 ! FORMAL PARAMETERS:
;  875          1054  1 !
;  876          1055  1 !       cirnamdsc       Descriptor of circuit name
;  877          1056  1 !
;  878          1057  1 !       cirblk          Address of longword in which to the return the
;  879          1058  1 !                       address of the circuit block if it is located
;  880          1059  1 !
;  881          1060  1 ! IMPLICIT INPUTS:
;  882          1061  1 !       cnf$gq_cirsurlst        List of circuits
;  883          1062  1 !
;  884          1063  1 ! ROUTINE VALUE:
;  885          1064  1 !
;  886          1065  1 !  True    Circuit block was found and address was returned in
;  887          1066  1 !          cirblk.
;  888          1067  1 !
;  889          1068  1 !  False   Circuit block was not found
;  890          1069  1 !--
;  891          1070  1
;  892          1071  2     BEGIN
;  893          1072  2     LOCAL
;  894          1073  2         CIRCUIT :       REF BBLOCK;
;  895          1074  2     MAP
;  896          1075  2         CIRNAMDSC :     REF BBLOCK;
;  897          1076  2
;  898          1077  2     CIRCUIT = .CNF$GQ_CIRSURLST;                    ! First circuit in list
;  899          1078  2     WHILE .CIRCUIT NEQ CNF$GQ_CIRSURLST DO          ! For all circuits in list
;  900          1079  3         BEGIN
;  901          1080  3         IF CH$EQL (.CIRCUIT [CIR$W_CIRNAMLEN], CIRCUIT [CIR$T_CIRNAM],
;  902          1081  3                    .CIRNAMDSC [DSC$W_LENGTH], .CIRNAMDSC [DSC$A_POINTER])
;  903          1082  3         THEN
;  904          1083  4             BEGIN
;  905          1084  4             .CIRBLK = .CIRCUIT;                     ! Return address of matching block
;  906          1085  4             RETURN TRUE;
;  907          1086  4             END
;  908          1087  3         ELSE
;  909          1088  3             CIRCUIT = .CIRCUIT [CIR$L_LINK];        ! Get next block
;  910          1089  2         END;                            ! WHILE traversing Circuit linked list
;  911          1090  2
;  912          1091  2     RETURN FALSE;
;  913          1092  1     END;                                ! Routine CNF$LOCATE_CIR_BLK


                                        003C 00000          .ENTRY  CNF$LOCATE_CIR_BLK, Save R2,R3,R4,R5        ; 1043
```

```
                                 54      0000G  CF  D0 00002          MOVL    CNF$GQ_CIRSURLST, CIRCUIT                          ; 1077
                                 55        04   AC  D0 00007          MOVL    CIRNAMDSC, R5                                      ; 1081
                                 50      0000G  CF  9E 00008 1$:       MOVAB   CNF$GQ_CIRSURLST, R0                               ; 1078
                                 50             54  D1 00010          CMPL    CIRCUIT, R0                                        ;
                                                19  13 00013          BEQL    3$                                                 ;
      04   BC          00    18  A4        16  A4  2D 00015          CMPC5   22(CIRCUIT), 24(CIRCUIT), #0, @CIRNAMDSC, -        ; 1080
                                             04  B5     0001D          @4(R5)                                                     ;
                                                08  12 0001F          BNEQ    2$                                                 ;
                                    08   BC          54  D0 00021          MOVL    CIRCUIT, @CIRBLK                               ; 1084
                                 50              01  D0 00025          MOVL    #1, R0                                             ; 1085
                                                     04 00028          RET                                                        ;
                                 54             64  D0 00029 2$:       MOVL    (CIRCUIT), CIRCUIT                             ; 1088
                                                DD  11 0002C          BRB     1$                                                 ; 1078
                                                50  D4 0002E 3$:       CLRL    R0                                             ; 1091
                                                    04 00030          RET                                                        ; 1092

; Routine Size:  49 bytes,    Routine Base:  $CODE$ + 04BA
```

```
;  915          1093  1 %SBTTL  'cnf_disable_surveillance  '
;  916          1094  1 ROUTINE CNF_DISABLE_SURVEILLANCE (IRB, KNOWN, CIRCUITNAM_DSC) =
;  917          1095  1
;  918          1096  1 !++
;  919          1097  1 !
;  920          1098  1 !  Jacket routine to ensure common error recovery and memory
;  921          1099  1 !  deallocation for the disabling of surveillance logic.
;  922          1100  1 !
;  923          1101  1 !       irb             Interrupt request block, containing request context
;  924          1102  1 !
;  925          1103  1 !       known           If true, then clear surveillance for all circuits
;  926          1104  1 !
;  927          1105  1 !       circuitnam_dsc  Descriptor for name of circuit to clear surveillance on.
;  928          1106  1 !
;  929          1107  1 !  Always return success, any errors are buffered and then sent to
;  930          1108  1 !  connectee.
;  931          1109  1 !--
;  932          1110  1
;  933          1111  2     BEGIN
;  934          1112  2     MAP
;  935          1113  2         CIRCUITNAM_DSC : REF BBLOCK;
;  936          1114  2     LOCAL
;  937          1115  2         CIRCUIT : REF BBLOCK,
;  938          1116  2         STATUS;
;  939          1117  2
;  940          1118  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;  941          1119  2         $DESCRIPTOR ('cnf_disable_surveillance'));
;  942          1120  2
;  943          1121  2     STATUS = DISABLE_SURVEILLANCE (.IRB, .KNOWN, .CIRCUITNAM_DSC);
;  944          1122  2     IF NOT .STATUS
;  945          1123  2     THEN
;  946          1124  2         CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_MPR, 0, .STATUS)
;  947          1125  2     ELSE
;  948          1126  2         CNF$BUFR_NICE_MSG (.IRB, SUCCESS_NICE_DSC, 0);
;  949          1127  2
;  950          1128  2     !
;  951          1129  2     !   Check to ensure that there is still something under surveillance,
;  952          1130  2     !   otherwise, clear flag so that when execution returns to primary loop
;  953          1131  2     !   in CNFMAIN it will terminate.
;  954          1132  2     !
;  955          1133  2     CNF$B_SURVEILLANCE_SET = FALSE;
;  956          1134  2     CIRCUIT = .CNF$GQ_CIRSURLST;
;  957          1135  2     WHILE .CIRCUIT NEQ CNF$GQ_CIRSURLST DO
;  958          1136  3         BEGIN
;  959          1137  3         IF .CIRCUIT [CIR$B_SURVEIL] EQL NMA$C_SUR_ENA
;  960          1138  3         THEN CNF$B_SURVEILLANCE_SET = TRUE;
;  961          1139  3         CIRCUIT = .CIRCUIT [CIR$L_LINK];
;  962          1140  2         END;                        ! WHILE traversing Circuit linked list
;  963          1141  2
;  964          1142  2     RETURN TRUE;
;  965          1143  1     END;                            ! Routine cnf_disable_surveillance


                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2
                                45  43  41  52  54  00104 P.AAX:  .ASCII  \TRACE\                                    ;
```

```
                                                00109                   .BLKB   3
                                       00000005 0010C P.AAW:            .LONG   5
                                       00000000' 00110                  .ADDRESS P.AAX
72 75 73 5F 65 6C 62 61 73 69 64 5F 66 6E 63   00114 P.AAZ:             .ASCII  \cnf_disable_surveillance\
                  65 63 6E 61 6C 6C 69 65 76   00123
                                       00000018 0012C P.AAY:            .LONG   24
                                       00000000' 00130                  .ADDRESS P.AAZ


                                                                        .PSECT  $CODE$,NOWRT,2

                                                0000 00000 CNF_DISABLE_SURVEILLANCE:
                                                                        .WORD   Save nothing                    ; 1094
                                       0000'  CF  9F 00002              PUSHAB  P.AAY                           ; 1119
                                       0000'  CF  9F 00006              PUSHAB  P.AAW                           ; 1118
                                              01  DD 0000A              PUSHL   #1
                                0000G  CF     03  FB 0000C              CALLS   #3, CNF$TRACE
                                       7E     08  AC 7D 00011          MOVQ    KNOWN, -(SP)                    ; 1121
                                              04  AC DD 00015          PUSHL   IRB
                                0000V  CF     03  FB 00018              CALLS   #3, DISABLE_SURVEILLANCE
                                       11         50 E8 0001D          BLBS    STATUS, 1$                      ; 1122
                                                  50 DD 00020          PUSHL   STATUS                          ; 1124
                                                  7E D4 00022          CLRL    -(SP)
                                       7E         05 CE 00024          MNEGL   #5, -(SP)
                                              04  AC DD 00027          PUSHL   IRB
                                0000G  CF     04  FB 0002A              CALLS   #4, CNF$BUFR_ERR_MSG
                                              0E  11 0002F              BRB     2$
                                       7E         D4 00031 1$:         CLRL    -(SP)                           ; 1126
                                       0000'  CF  9F 00033              PUSHAB  SUCCESS_NICE_DSC
                                              04  AC DD 00037          PUSHL   IRB
                                0000G  CF     03  FB 0003A              CALLS   #3, CNF$BUFR_NICE_MSG
                                       0000G  CF  D4 0003F 2$:         CLRL    CNF$B_SURVEILLANCE_SET          ; 1133
                                       0000G  CF  D0 00043              MOVL    CNF$GQ_CIRSURLST, CIRCUIT       ; 1134
                                       0000G  CF  9E 00048 3$:          MOVAB   CNF$GQ_CIRSURLST, R0            ; 1135
                                              51  D1 0004D              CMPL    CIRCUIT, R0
                                              0F  13 00050              BEQL    5$
                                              0A  A1 95 00052          TSTB    10(CIRCUIT)                     ; 1137
                                                  05 12 00055          BNEQ    4$
                                0000G  CF     01  D0 00057              MOVL    #1, CNF$B_SURVEILLANCE_SET      ; 1138
                                       51         61 D0 0005C 4$:      MOVL    (CIRCUIT), CIRCUIT              ; 1139
                                                  E7 11 0005F          BRB     3$                              ; 1135
                                       50         01 D0 00061 5$:      MOVL    #1, R0                          ; 1142
                                                     04 00064          RET                                     ; 1143

; Routine Size:  101 bytes,    Routine Base:  $CODE$ + 04EB
```

```
;  967          1144  1 %SBTTL  'disable_surveillance  '
;  968          1145  1 ROUTINE DISABLE_SURVEILLANCE (IRB, KNOWN, CIRCUITNAM_DSC) =
;  969          1146  1 !++
;  970          1147  1 !
;  971          1148  1 !  Perform some checking before calling the routine which will
;  972          1149  1 !  handle the actual disabling of surveillance on a circuit by
;  973          1150  1 !  first determining if the requested circuit has surveillance set.
;  974          1151  1 !  If known was specified, then discover all the NI circuits available.
;  975          1152  1 !
;  976          1153  1 !       irb             Interrupt request block, containing request context
;  977          1154  1 !
;  978          1155  1 !       known           If true, then clear surveillance for all circuits
;  979          1156  1 !
;  980          1157  1 !       circuitnam_dsc  For checking if this circuit is in our list
;  981          1158  1 !                       of circuits.
;  982          1159  1 !--
;  983          1160  2     BEGIN
;  984          1161  2     LOCAL
;  985          1162  2         CIRCUIT :       REF BBLOCK;
;  986          1163  2     MAP
;  987          1164  2         CIRCUITNAM_DSC : REF BBLOCK;
;  988          1165  2
;  989          1166  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;  990          1167  2         $DESCRIPTOR ('disable_surveillance'));
;  991          1168  2
;  992          1169  2     IF .KNOWN
;  993          1170  2     THEN
;  994          1171  2         !
;  995          1172  2         !   For every circuit in the list, disable surveillance
;  996          1173  2         !
;  997          1174  3         BEGIN
;  998          1175  3         CIRCUIT = .CNF$GQ_CIRSURLST;
;  999          1176  3         WHILE .CIRCUIT NEQ CNF$GQ_CIRSURLST DO
; 1000          1177  4             BEGIN
; 1001          1178  4             EXECUTE (CNF$DISABLE_SUPVEIL (.CIRCUIT));
; 1002          1179  4             CIRCUIT = .CIRCUIT [CIR$L_LINK];
; 1003          1180  3             END;                            ! WHILE traversing circuit linked list
; 1004          1181  3         END
; 1005          1182  2     ELSE
; 1006          1183  2         !
; 1007          1184  2         !   If the circuit is in our list, then disable surveillance,
; 1008          1185  2         !   otherwise buffer an error for return to connectee.
; 1009          1186  2         !
; 1010          1187  3         BEGIN
; 1011          1188  3         IF CNF$LOCATE_CIR_BLK (.CIRCUITNAM_DSC, CIRCUIT)
; 1012          1189  3         THEN
; 1013          1190  4             EXECUTE (CNF$DISABLE_SURVEIL (.CIRCUIT))
; 1014          1191  3         ELSE
; 1015          1192  4             BEGIN                           ! This circuit not in data base
; 1016          1193  4             CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_IDE, NMA$C_ENT_CIR, 0,
; 1017          1194  4                 .CIRCUITNAM_DSC);
; 1018          1195  4             RETURN TRUE;
; 1019          1196  3             END;
; 1020          1197  2         END;
; 1021          1198  2
; 1022          1199  2     RETURN TRUE;
; 1023          1200  1     END;                            ! Routine disable_surveillance
```

```
                                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                45  43  41  52  54  00134 P.ABB:  .ASCII  \TRACE\                                    ;
                                                                    00139        .BLKB   3                                            ;
                                                        00000005    0013C P.ABA:  .LONG   5                                            ;
                                                        00000000'   00140        .ADDRESS P.ABB                                       ;
6C  69  65  76  72  75  73  5F  65  6C  62  61  73  69  64  00144 P.ABD:  .ASCII  \disable_surveillance\                               ;
                                                65  63  6E  61  6C  00153                                                           ;
                                                        00000014    00158 P.ABC:  .LONG   20                                           ;
                                                        00000000'   0015C        .ADDRESS P.ABD                                       ;


                                                                                .PSECT  $CODE$,NOWRT,2

                                                            0000 00000 DISABLE_SURVEILLANCE:
                                                                                .WORD   Save nothing                                 ; 1145
                                            5E                  04  C2 00002           SUBL2   #4, SP                               ;
                                                    0000'       CF  9F 00005           PUSHAB  P.ABC                                ; 1167
                                                    0000'       CF  9F 00009           PUSHAB  P.ABA                                ; 1166
                                                                01  DD 0000D           PUSHL   #1                                   ;
                                        0000G       CF              03  FB 0000F       CALLS   #3, CNF$TRACE                        ;
                                            1D          08          AC  E9 00014       BLBC    KNOWN, 2$                            ; 1169
                                            6E      0000G       CF  D0  00018       MOVL    CNF$GQ_CIRSURLST, CIRCUIT            ; 1175
                                            50      0000G       CF  9E  0001D 1$:   MOVAB   CNF$GQ_CIRSURLST, R0                 ; 1176
                                            50                  6E  D1  00022       CMPL    CIRCUIT, R0                          ;
                                                                37  13  00025       BEQL    4$                                   ;
                                                                6E  DD  00027       PUSHL   CIRCUIT                              ; 1178
                                        0000V       CF              01  FB 00029       CALLS   #1, CNF$DISABLE_SURVEIL              ;
                                            30                  50  E9  0002E       BLBC    STATUS, 5$                           ;
                                                                9E  DD  00031       PUSHL   @CIRCUIT                             ; 1179
                                                                E8  11  00033       BRB     1$                                   ; 1176
                                                                5E  DD  00035 2$:   PUSHL   SP                                   ; 1188
                                                        0C      AC  DD  00037       PUSHL   CIRCUITNAM_DSC                       ;
                                        FF2B        CF              02  FB 0003A       CALLS   #2, CNF$LOCATE_CIR_BLK               ;
                                            0B                  50  E9  0003F       BLBC    R0, 3$                               ;
                                                                6E  DD  00042       PUSHL   CIRCUIT                              ; 1190
                                        0000V       CF              01  FB 00044       CALLS   #1, CNF$DISABLE_SURVEIL              ;
                                            12                  50  E8  00049       BLBS    STATUS, 4$                           ;
                                                                    04  0004C       RET                                          ;
                                                        0C      AC  DD  0004D 3$:   PUSHL   CIRCUITNAM_DSC                       ; 1194
                                                7E              03  7D  00050       MOVQ    #3, -(SP)                            ; 1193
                                                7E              09  CE  00053       MNEGL   #9, -(SP)                            ;
                                                        04      AC  DD  00056       PUSHL   IRB                                  ;
                                        0000G       CF              05  FB 00059       CALLS   #5, CNF$BUFR_ERR_MSG                 ;
                                                50                  01  D0  0005E 4$:   MOVL    #1, R0                               ; 1199
                                                                    04  00061 5$:   RET                                          ; 1200

; Routine Size:  98 bytes,    Routine Base:  $CODE$ + 0550
```

```
;  1025     1201  1 %SBTTL  'CNF$DISABLE_SURVEIL:  clean up circuit block entry and quit surveillance'
;  1026     1202  1 GLOBAL ROUTINE CNF$DISABLE_SURVEIL (CIR) =
;  1027     1203  1
;  1028     1204  1 !++
;  1029     1205  1 ! FUNCTIONAL DESCRIPTION:
;  1030     1206  1 !
;  1031     1207  1 !  This is the routine that actually terminates surveillance of a circuit.
;  1032     1208  1 !
;  1033     1209  1 ! FORMAL PARAMETERS:
;  1034     1210  1 !
;  1035     1211  1 !       cir     Circuit control block.
;  1036     1212  1 !
;  1037     1213  1 ! IMPLICIT INPUTS:
;  1038     1214  1 !       NONE
;  1039     1215  1 !
;  1040     1216  1 ! IMPLICIT OUTPUTS:
;  1041     1217  1 !       NONE
;  1042     1218  1 !
;  1043     1219  1 ! ROUTINE VALUE:
;  1044     1220  1 ! COMPLETION CODES:
;  1045     1221  1 !       NONE
;  1046     1222  1 !
;  1047     1223  1 ! SIDE EFFECTS:
;  1048     1224  1 !       NONE
;  1049     1225  1 !
;  1050     1226  1 !--
;  1051     1227  1
;  1052     1228  2     BEGIN
;  1053     1229  2     MAP
;  1054     1230  2         CIR : REF BBLOCK;
;  1055     1231  2     LOCAL
;  1056     1232  2         SID : REF BBLOCK,
;  1057     1233  2         STATUS;
;  1058     1234  2
;  1059     1235  2
;  1060     1236  2     CIR [CIR$B_SURVEIL] = NMA$C_SUR_DIS;        ! Mark surveillance disabled
;  1061     1237  2
;  1062     1238  2     EXECUTE ( $DASSGN (CHAN = .CIR [CIR$W_CHAN]) );     ! Terminate read of System ID's
;  1063     1239  2
;  1064     1240  2     !
;  1065     1241  2     !   Deallocate all the memory used to store system ID messages
;  1066     1242  2     !   gathered for the circuit
;  1067     1243  2     !
;  1068     1244  2     SID = .CIR [CIR$L_SIDFLINK];
;  1069     1245  2     WHILE .SID NEQ CIR [CIR$L_SIDFLINK] DO
;  1070     1246  3         BEGIN
;  1071     1247  3         REMQUE (.SID, STATUS);
;  1072     1248  3         EXECUTE (CNF$FREE_VM (%REF(SID$C_LENGTH), SID));
;  1073     1249  3         SID = .CIR [CIR$L_SIDFLINK];
;  1074     1250  3         END;
;  1075     1251  2
;  1076     1252  2     !
;  1077     1253  2     !   Record time when surveillance was discontinued
;  1078     1254  2     !
;  1079     1255  2     EXECUTE ($GETTIM (TIMADR = CIR [CIR$Q_ELAPSDTIM]) );
;  1080     1256  2
;  1081     1257  2     RETURN TRUE;
```

```
; 1082          1258  1     END;                            ! Routine cnf$disable_surveil


                                                                        .EXTRN  SYS$DASSGN

                                                    000C 00000          .ENTRY  CNF$DISABLE_SURVEIL, Save R2,R3                 ; 1202
                                                 08 C2 00002            SUBL2   #8, SP                                          ;
                                              5E    52 00005            MOVL    CIR, R2                                         ; 1236
                                              52 04 AC D0 00009            MOVB    #1, 10(R2)                                      ;
                                           0A    A2 01 90 0000D            CVTWL   20(R2), -(SP)                                   ; 1238
                                           7E 14 A2 32 00011            CALLS   #1, SYS$DASSGN                                  ;
                                 00000000G 00    01 FB 00018            BLBC    STATUS, 3$                                      ;
                                           4B    50 E9 0001B            MOVL    64(R2), SID                                     ;
                                        04 AE 40 A2 D0 00020            ADDL3   #64, CIR, R2                                    ; 1244
                                     52 04 AC 00000040 8F C1 00029 1$:  CMPL    SID, R2                                         ; 1245
                                           52 04 AE D1 00030            BEQL    2$                                              ;
                                              25 13 00032               REMQUE  @SID, STATUS                                    ; 1247
                                           53 04 BE 0F 00034            PUSHAB  SID                                             ; 1248
                                              04 AE 9F 00038            MOVL    #37, 4(SP)                                      ;
                                           04 AE 25 D0 0003B            PUSHAB  4(SP)                                           ;
                                        0000G CF    04 FB 0003F         CALLS   #2, CNF$FREE_VM                                 ;
                                              21    50 E9 00042         BLBC    STATUS, 3$                                      ;
                                     52 04 AC 00000040 8F C1 00045      ADDL3   #64, CIR, R2                                    ; 1249
                                        04 AE          62 D0 0004E      MOVL    (R2), SID                                       ;
                                                       D5 11 00052      BRB     1$                                              ; 1245
                                     7E 04 AC          30 C1 00054 2$:  ADDL3   #48, CIR, -(SP)                                 ; 1255
                                 00000000G 00          01 FB 00059      CALLS   #1, SYS$GETTIM                                  ;
                                           03          50 E9 00060      BLBC    STATUS, 3$                                      ;
                                           50          01 D0 00063      MOVL    #1, R0                                          ; 1257
                                                       04 00066 3$:     RET                                                     ; 1258

; Routine Size:  103 bytes,    Routine Base:  $CODE$ + 05B2
```

```
; 1084        1259  1 END                          ! End of module CNFREQUES
; 1085        1260  0 ELUDOM


                                                    .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;   $PLIT$                        352  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                          80  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                       1561  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                           Library Statistics
;                                        --------- Symbols ---------      Pages       Processing
;       File                             Total     Loaded    Percent      Mapped      Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1     9776         15          0         581        00:01.0
;   _$255$DUA28:[SHRLIB]NET.L32;1         1279         16          1          63        00:00.8
;   _$255$DUA28:[SHRLIB]NMALIBRY.L32;1     887         37          4          47        00:00.8



;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CNFREQUES/OBJ=OBJ$:CNFREQUES MSRC$:CNFREQUES/UPDATE=(ENH$:CNFREQUES)

; Size:          1561 code + 432 data bytes
; Run Time:        00:33.3
; Elapsed Time:    01:01.0
; Lines/CPU Min:    2268
; Lexemes/CPU-Min: 18605
; Memory Used:  213 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NICNF

CNFDEF
SDL

CNFDEF
LIS

NETTRN
LIS

NICONFIG
MAP

NETTREE
LIS

CNFMAIN
LIS

CNFREQUES
LIS

SERVER
LIS

CNFINTRPT
LIS

CNFWQDEF
SDL

CNFPREFIX
REQ

CNFMSG
LIS

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CNFSEND
LIS

CNFSTORE
LIS

NML.SHR
MAP

CNFSHOW
LIS

CNFWQDEF
LIS

NML
MAP

CNFWORKQ
LIS

NPADEF
MDL

NMAHEAD
B32

NPAMAC
MAR

NMATAIL
B32

NMLDDL
B32

NMLDEF
MDL

NMAFILES
LIS

NMAFIELDS
LIS